Practical Workbook
COMPUTER COMMUNICATION NETWORKS

Name : ______________________

Year : ______________________

Batch : ______________________

Roll No : ______________________

Department: ______________________

Department of Computer & Information Systems Engineering

NED University of Engineering & Technology,
Karachi – 75270, Pakistan

Practical Workbook
COMPUTER COMMUNICATION NETWORKS

Prepared by:
Ms. Sumayya Zafar
Ms. Ibshar Ishrat

Revised in:
February 2019

Department of Computer & Information Systems Engineering
NED University of Engineering & Technology,
Karachi – 75270, Pakistan

INTRODUCTION

This workbook has been compiled to assist the conduct of practical classes for CS-318 Computer Communication Networks. Practical work relevant to this course aims at providing students a good grasp on the various topics in computer networks. This laboratory manual aims to augment the classroom teaching of the course and to provide the students essential practical knowledge in the subject.

Each component of undergraduate engineering curriculum is accompanied with some learning outcomes for students. Consequently, the practical portion of CS-318 Computer Communication Networks is designed to help students "**practice** configuration and troubleshooting of computer networks using modern tools. The exercises are designed to enhance cognitive (application level) as well as psychomotor (guided-response level) skills, enabling the students for Modern Tool Usage. A rubric to evaluate student performance has been provided at the end of the workbook.

The first section of manual deals with configuration of IP Addresses, making network cables, establishing connectivity between devices & exploring Cisco IOS commands. The second section covers routing protocols: static and dynamic, data-link layer protocol and access lists configuration. The third section covers basic LAN switch operation, loop avoidance using Spanning Tree Protocol and Virtual LANs. The final section is based on advanced topics of NAT, BGP, DHCP and Wireshark.

This workbook is designed to assist both instructor and student, practically realizing theoretical concepts of the course, providing in-depth understanding and elaborate various areas of computer networks.

CONTENTS

Lab Session 01

OBJECT

Configure IP Addresses on different network devices using Packet Tracer

THEORY

In this lab session, you will be getting familiar with basic network devices and how to connect and configure them.

Packet Tracer is a cross-platform visual simulation tool designed by Cisco Systems that allows users to create network topologies and imitate modern computer networks. The software allows users to simulate the configuration of Cisco routers and switches using a simulated command line interface.

Network Devices

Switch

Switches allow different devices in a network to communicate. They operate at Layer 2 of OSI model. Data is transmitted in the form of frames. It is used in LAN (Local Area Network). Switches use MAC address for data transmission. More details about switching operation are covered in Lab Session 09 of this manual. Cisco has a comprehensive portfolio of network switches.

Router

Routers allow different networks to communicate. They operate at Layer 3 of OSI model. As the name suggests, they route data packets between computer networks. A router can connect networked computers to the Internet, so multiple users can share a connection. Routers help connect networks within an organization or connect the networks of multiple branch locations. Data is transmitted in the form of packets. It is used in LAN, MAN and WAN. A router uses IP address for data transmission.

An IP (Internet Protocol) address uniquely identifies a node or host connection to an IP network. System administrators or network designers assign IP addresses to nodes. IP addresses are configured by software and are not hardware specific. An IP address is a 32-bit binary number usually represented as four fields each representing 8-bit numbers in the range 0 to 255 (sometimes called octets) separated by decimal points.
For example: 150.215.17.9

It is sometimes useful to view the values in their binary form.

150.215.17.9
10010110.11010111.00010001.00001001
For more details on IP address, refer to Appendix A.

Some Basic Commands to Interact with Cisco Routers

Cisco routers support different modes of operation. When you access a router, it will typically be in the “user” mode. User mode gives a user access to simple “show commands.”

From user mode the next step is “Privileged mode.” In the “Privileged mode” a user can have full access to all the databases maintained by the router. Cisco routers use many other modes, but let us keep it simple for now.

Configuration mode
Privileged mode
User mode

Let’s connect the devices and configure IP addresses. Consider the following topology.

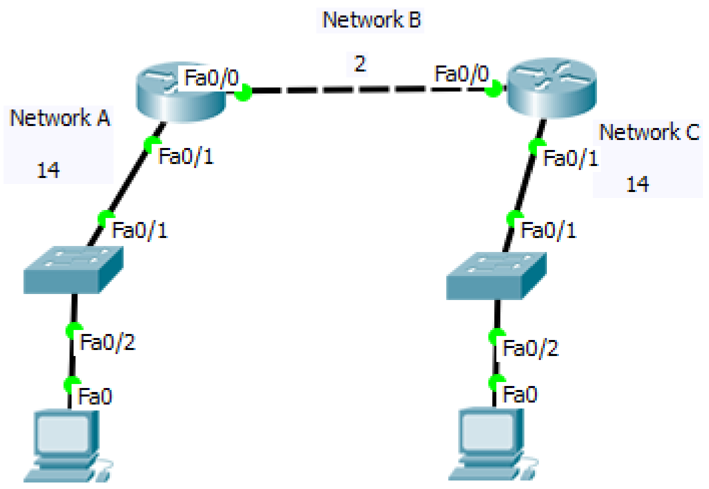


Figure 1.1: A simple network topology

1- Select router 1841. Connect a 24-port switch with the router using straight-through cable. (Cables would be discussed in the next lab session)

2- Select a generic PC (end device) connect it with a port on switch using straight through cable.

3- Repeat steps 1 and 2. Connect both the routers using a copper cross-over cable.

4- Subnet the network address 192.168.10.0/24 such that there are 14 usable hosts in n/w A and n/w C, and 2 usable hosts between the routers i.e. n/w B in the above figure. (Refer to appendix A)

5- Find first host, last host and broadcast address for each network. Assign any two host addresses to the interfaces between the devices.

6- Double click on the router. Press “enter” key. You should get a prompt that looks like router>. You are now in “user mode”.

7- To enter into the “Privileged Mode” type “`enable`” or “`en`”. The prompt should end with# e.g. (`Router#`). Type “?” to view the commands available in this mode.

8- Now let’s move to “Configuration Mode”. Type “`configure terminal”.` This will take you to configuration mode. The prompt ends with (`router-config`) #
Now type these commands:

```
config t
```

```
interface fa0/1
```

This puts you in interface mode. Now you can configure interface fa0/1.

```
e.g. ip address 192.168.10.1 255.255.255.0
```

This gives the interface an IP address and subnet mask.

```
no shutdown
```

9- Repeat the same for other interfaces of routers.

EXERCISE

1. Given a Class C network address 192.168.20.0 (/24). Divide it into three sub-networks each with unequal number of hosts’ requirement as shown below:
   Subnet A: 90 Hosts, Subnet B: 2 Hosts, Subnet C: 23 Hosts. Simulate a simple network topology and configure IP address on interfaces. Also label them. Attach screenshot of topology.

Attach screenshot here

Lab Session 02

OBJECT

Practice making the following kinds of UTP cables:

1. ***Straight through cable***
2. ***Cross cable***

THEORY

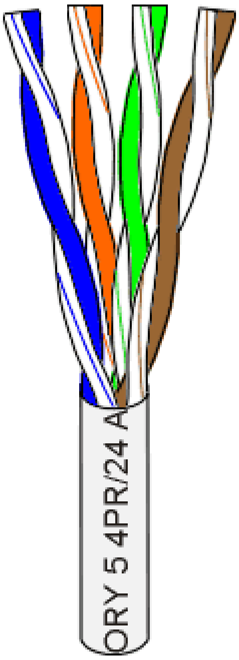


Figure 2.1: UTP cable

There are several classifications of twisted pair cable. Let's skip right over them and state that we'll use Category 5 (or CAT 5) cable for all new installations. Likewise, there are several fire code classifications for the outer insulation of CAT 5 cable. We'll use CMR cable, or "riser cable," for most of the wiring we do. You should also be aware of CMP or plenum cable (a plenum is used to distribute air in a building) you may be required by local or national codes to use the more expensive plenum-jacketed cable if it runs through suspended ceilings, ducts, or other areas, if they are used to circulate air or act as an air passage from one room to another. If in doubt, use plenum. CMR cable is generally acceptable for all applications not requiring plenum cable.

CAT 5 cable is available in reel-in-box packaging. This is very handy for pulling the wire without putting twists in it. Without this kind of package or a cable reel stand, pulling wire is a two-person job. Before the advent of the reel-in-box, we used to put a reel of wire on a broom handle to pull it. One person would hold the broom handle and the other would pull broom handle to pull it. You will produce a tangled mess, if your pull the wire off the end of the reel alone.

Standard wire patch cables are often specified for cable segments running form a wall jack to a PC and for patch panels. They are more flexible than solid core wire. However, the rationale for using it is that the constant flexing of patch cables may wear-out solid core cable and break it. This is not a real concern in the average small network.

Most of the wiring we do simply connects computers directly to other computers or hubs. Solid core cable is quite suitable for this purpose and for many home and small business network. It is also quite acceptable for use as patch cables. You might consider a stranded wire patch cable, if you have a notebook computer you are constantly moving around.

CAT 5 cable has four twisted pairs of wire for a total of eight individually insulated wires. Each pair is color coded with one wire having solid color (blue, orange, green, or brown) twisted around a second wire with a white background and a stripe of the same color. The solid color may have white stripe in some cables. Cable colors are commonly described using the background color followed by the color of the stripe; e.g.; white-orange is a wire with a white background and an orange stripe.

Connectors

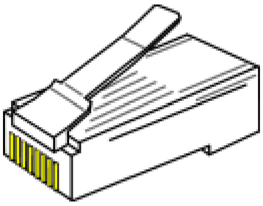


Figure 2.2: RJ-45 Connector

The straight through and cross-over patch cables are discussed in this article, which are terminated with CAT 5 RJ-45 modular plugs. RJ-45 plugs are similar to those you'll see on the end of your telephone cable except they have eight as opposed to four or six contacts on the end of the plug and they are about twice as big. Make sure they are rated for CAT 5 wiring. (RJ stands for "Registered Jack"). Also, there are RJ-45 plugs designed for both solid core wire and stranded wire. Others are designed specifically for one kind of wire or the other. Be sure you buy plugs appropriate for the wire you are going to use. We normally use plugs designed to accommodate both kinds of wire.

Network cabling tools

1. Modular Plug Crimp Tool

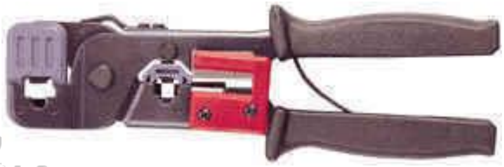


Figure 2.3: Modular plug crimp tool

You will need a modular crimp tool. This is very similar to the ones which have been used for many years for all kinds of telephone cable work and it works just fine for Ethernet cables. You don't need a lot of bells and whistles, just a tool which will securely crimp RJ-45 connectors. Some crimpers have cutters which can be used to cut the cable and individual wires, and possibly stripping the outer jacket.

2. Universal UTP Stripping Tool (Eclipse)

It makes a much neater cut. It is highly recommending for anyone who will make a lot of cables.

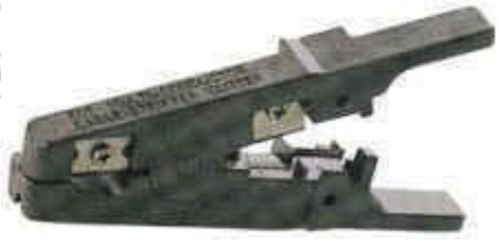


Figure 2.4: Eclipse

3. Diagonal Cutters

It is easier to use diagonal cutters ("diags" or "dikes") to cut the cable off at the reel and to fine-tune the cable ends during assembly. Also, if you don't have a stripper, you can strip the cable by using a small knife to carefully slice the outer jacket longitudinally and use the diags to cut it off around the circumference.

Figure 2.5: Diagonal cutters

UTP basics

The 10BASE-T and 100BASE-TX Ethernet consist of two transmission lines. Each transmission line is a pair of twisted wires. One pair receives data signals and the other pair transmits data signals. A balanced line driver or transmitter is at one end of one of these lines and a line receiver is at the other end. A (much) simplified schematic for one of these lines and its transmitter and receiver follows:

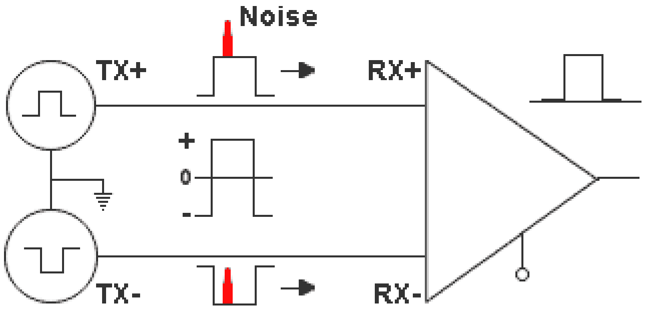


Figure 2.6: Schematic diagram of transmission line

Pulses of energy travel down the transmission line at about the speed of light (186,000 miles/second). The principal components of these pulses of energy are the potential difference between the wires and the current flowing near the surface of the wires. This energy can also be considered as residing in the magnetic field which surrounds the wires and the electric field between the wires. In other words, an electromagnetic wave which is guided by, and travels down the wires.

The main concern are the transient magnetic fields which surround the wires and the magnetic fields generated externally by the other transmission lines in the cable, other network cables, electric motors, fluorescent lights, telephone and electric lines, lightning, which may literally bury the Ethernet pulses, the conveyor of the information being sent down the line.

The twisted-pair Ethernet employs two principal means for combating noise. The first is the use of balanced transmitters and receivers. A signal pulse actually consists of two simultaneous pulses relative to ground: a negative pulse on one line and a positive pulse on the other. The receiver detects the total difference between these two pules. Since a pulse of noise usually produces pulses of the same polarity on both lines, it is essentially canceled out at the receiver. Also, the magnetic field surrounding one wire from a signal pulse is a mirror of the one on the other wire. At a very short distance from the two wires the magnetic fields are opposite and have a tendency to cancel the effect of each other out. This reduces the line's impact on the other pairs of wires and the rest of the world.

The second and the primary means of reducing cross-talk (the term cross-talk came from the ability to overhear conversations on other lines on your phone) between the pairs in the cable, is the double helix configuration produced by twisting the wires together. This configuration produces symmetrical (dentinal) noise signals in each wire. Ideally, their difference as detected at the receiver is zero. In actuality it is much reduced.

Straight through and cross over cable

Again, the wire with colored backgrounds may have white stripes and may be denoted that way in diagrams found elsewhere. For example, the green wire may be labeled Green-White. The background color is always specified first.

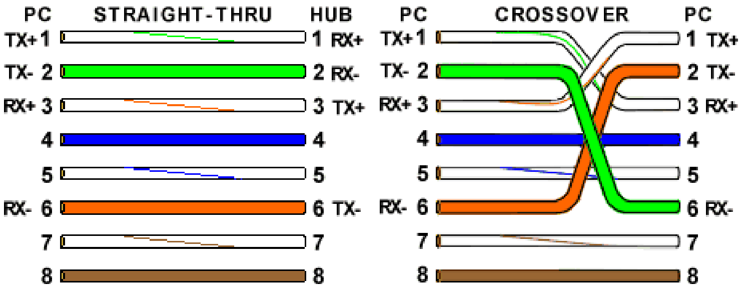


Figure 2.7: Straight through and crossover cable wire scheme

A Straight-through cable has identical ends, whereas a Crossover cable has different ends.

EIA/TIA 568A and 568B standards

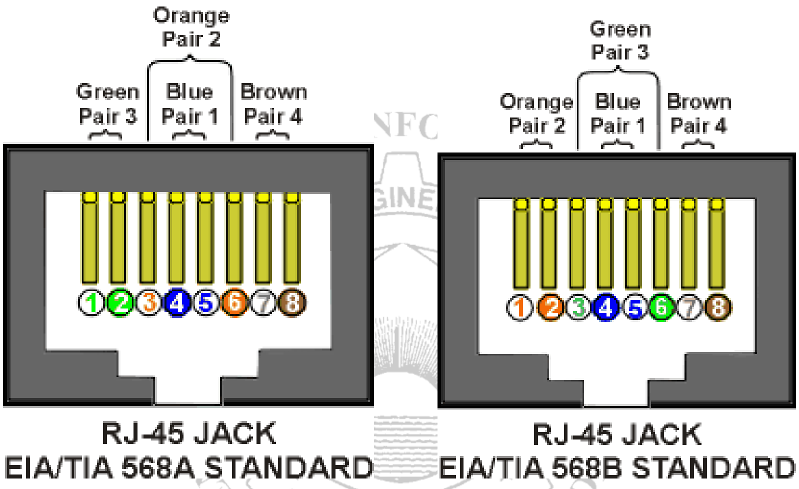


Figure 2.8: Cable connector standard ordering

It makes no functional difference which standard you use for a straight-through cable. You can start a crossover cable with either standard as long as the other end is the other standard. It makes no functional difference which end is which. Despite what you may have read elsewhere, a 568A patch cable will work in a network with 568B wiring and 568B patch cable will work in a 568A network. The electrons couldn't care less.

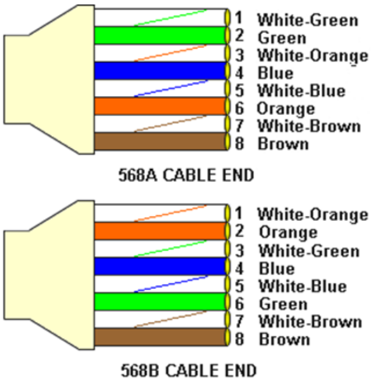


Figure 2.9: EIA/TIA 568A and 568B

PROCEDURE

To Make Cable

1. Pull the cable off the reel to the desired length and cut the total length of wire segments between a PC and a hub or between two PC's cannot exceed 100 Meters (328 feet or about the length of a football field) for 100BASE-TX and 300 Meters for 100BASE-T.

2. Strip one end of the cable with the stripper or a knife and diags. If you are using the stripper, place the cable in the groove on the blade (left) side of the stripper and align the end of the cable with the right side of the stripper. This will strip about ½” of the jacket off the cable. Turn the stripper about 1 ¼ turn and pull. If you turn it more, you will probably nick the wires. If you are using knife and diags, carefully slit the cable for about an inch or so and neatly trim around the circumference of the cable with diags to remove the jacket.

3. Inspect the wires for nicks. Cut off the end and start over if you see any. You may have to adjust the blade with the screw at the front stripper. Cable diameters and jacket thicknesses vary.

4. Spread and arrange the pairs roughly in the order of the desired cable end.

5. Untwist the pairs and arrange the wires in the order of the desired cable end. Flatten the end between your thumb and forefinger. Trim the ends of the wires so they are even with one another.

   It is very important that the unstripped (untwisted) end be slightly less than ½” long. If it is longer than ½” it will be out-of-spec and susceptible to crosstalk. If it is less than ½” it will not be properly clinched when RJ-45 plug is crimped on. Flatten again. There should be little or no space between the wires.

6. Hold the RJ-45 plug with the clip facing down or away from you. Push the wire firmly into the plug. **Now, inspect before crimping and wasting the plug!** Looking through the bottom of the plug, the wire on the far-left side will have a white background. The wires should alternative light and dark from left to right. The furthest right wire is brown. The wires should all end evenly at the front of the plug. The jacket should end just about where you see it in the diagram-right on the line.

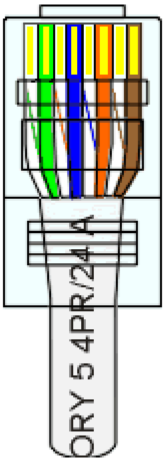


Figure 2.10:
Preparing the RJ-45 Connector

ALL ABOUT CRIMPING

7. Hold the wire near the RJ-45 plug with the clip down and firmly push it into the left side of the front of the Crimper (it will only go in one way). Hold the wire in place and squeeze the crimper handles quite firmly. This is what will happen:

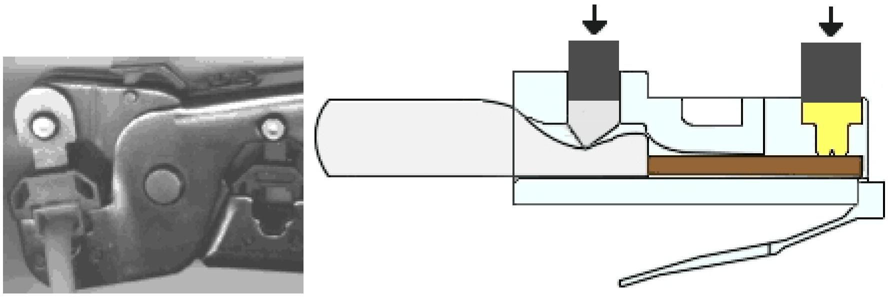


Figure 2.11: Crimping

(Crimp it once). The crimper pushes two plungers down on the RJ-45 plug. One forces, what amounts to, a cleverly designed plastic plug/wedge onto the cable jacket and very firmly clinches it. The other seats the “pins”, each with two teeth at its end, through the insulation and into the conductors of their respective wires.

8. Test the crimp… if done properly an average person will not be able to pull the plug off the cable with his or her bare hands. And that quite simply, besides lower cost, is the primary advantage of twisted-pair cables over the older thin wire, coaxial cables. In fact, the ease of installation and the modular RJ-45 plug is the main reason coaxial cable is no longer widely used for small Ethernet. But, don’t pull that hard on the plug. It could stretch the cable and change its characteristics. Look at the side of the plug and see if it looks like the diagram and give it a fairly firm tug to make sure it is crimped well.

9. Prepare the other end of the cable so it has the desired end and crimp.

10. If both ends of the cable are within reach, hold them next to each other and with RJ-45 clips facing away. Look through the bottom of the plugs. If the plugs are wired correctly, and they are identical, it is a straight-through cable. If they are wired correctly and they are different, it is a crossover cable.

PRECAUTIONS

1. Try to avoid running cables parallel to power cables.

2. If you bundle a group of cables together with cable ties (zip ties), do not over-clinch them. It is okay to snug them together firmly; but don’t tighten them so much that you deform the cables.

3. Keep cables away from devices which can introduce noise into them. Here’s a short list: electric heaters, loud speakers, printers, TV sets, fluorescent light, copiers, welding machines, microwave ovens, telephones, fans, elevator motors, electric ovens, dryers, washing machines, and shop equipment.

4. Avoid stretching UTP cables (the force should not exceed 24 LBS).

5. Do not use a stapler to secure UTP cables. Use telephone wire hangers, which are available at most hardware stores.

EXERCISES

1. Practice making both straight through and cross over cables. Test you cable using the lab tester available in lab. Write down your result for both cables.

____

____

____

____

____

2. Connect your home PC and router using cross-over cable. Verify the connectivity.

Lab Session 03

OBJECT

Configure File Sharing Server across different networks

THEORY

Welcome to “hands on routing.” The goal of this lab is to introduce you to Cisco devices and configure a simple file sharing server across different networks. Let’s consider the topology we simulated in Lab Session 01.

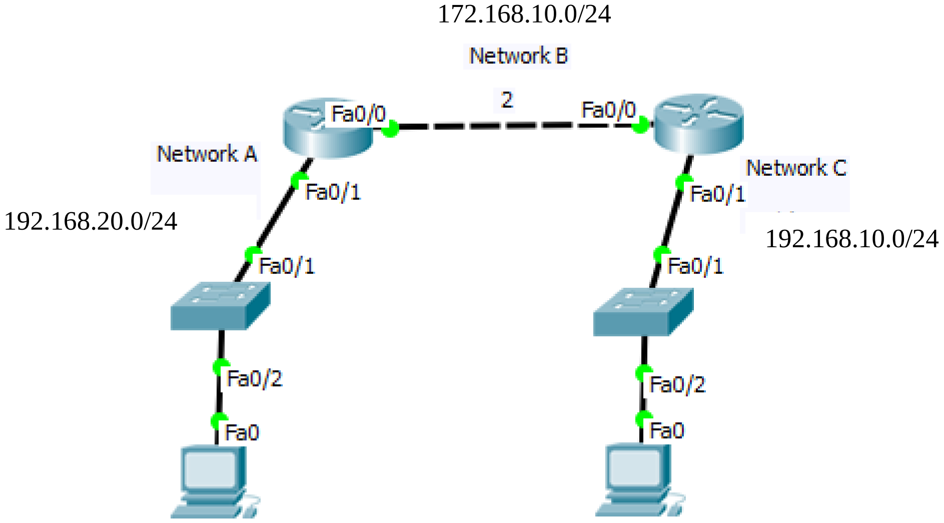
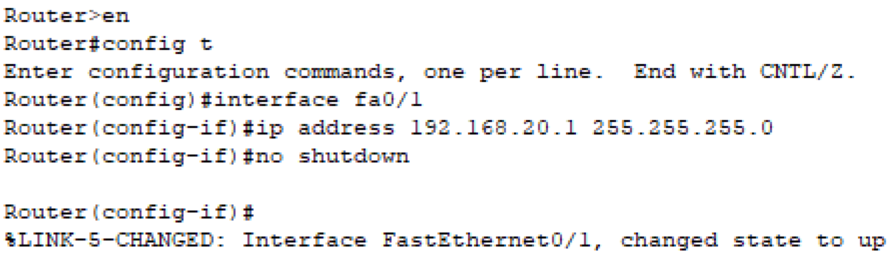


Figure 3.1: Scenario to configure file sharing server across different networks

PROCEDURE

1- Connect your PC with a 24-port switch using the straight through cable you made in the previous session.
2- Connect one port of the switch with the router using another straight through cable.
3- Configure IP address on interface fa0/1.



```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fa0/1
Router(config-if)#ip address 192.168.20.1 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up
```

4- Now configure IP address and gateway for your computer. On the lower right of your screen, right click on network—Open Network & Internet settings:

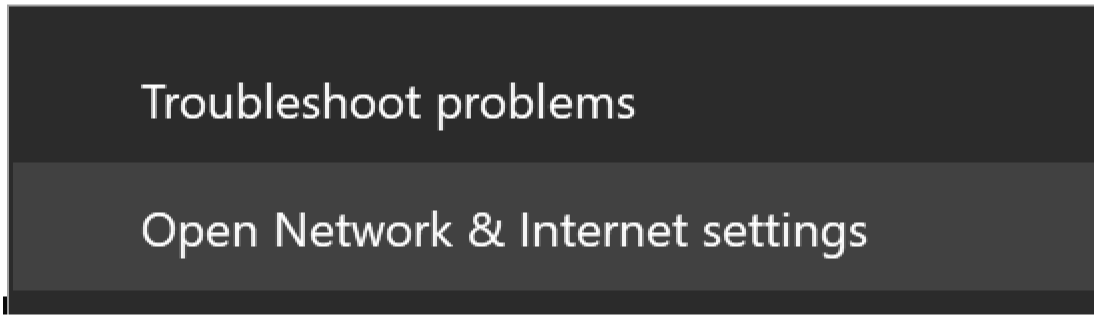


Figure 3.2: Network & Internet settings

5- Click Ethernet on the left, then on the right, under Related Settings, click on change adapter options:

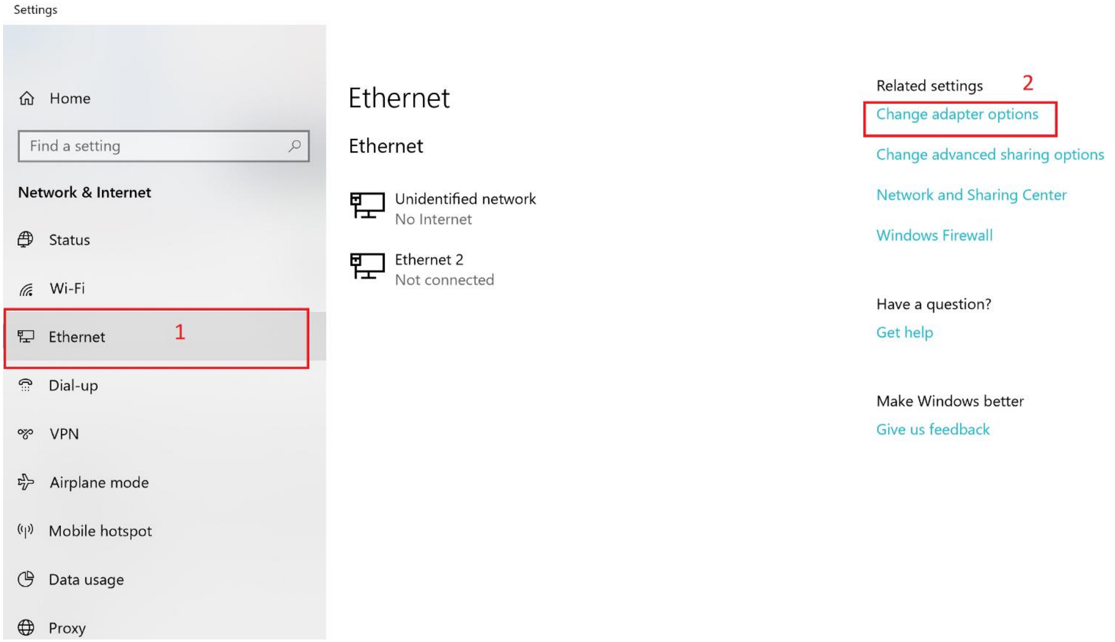


Figure 3.3: Change adapter options

6- Find the Ethernet connection in the Network Connections.

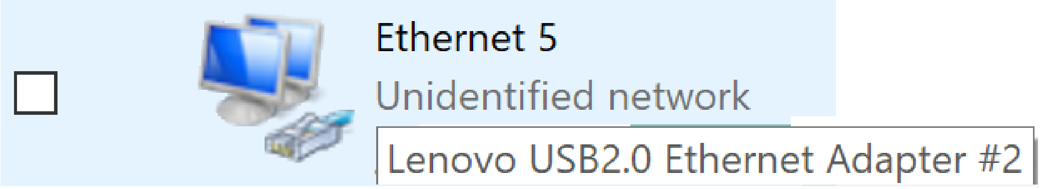
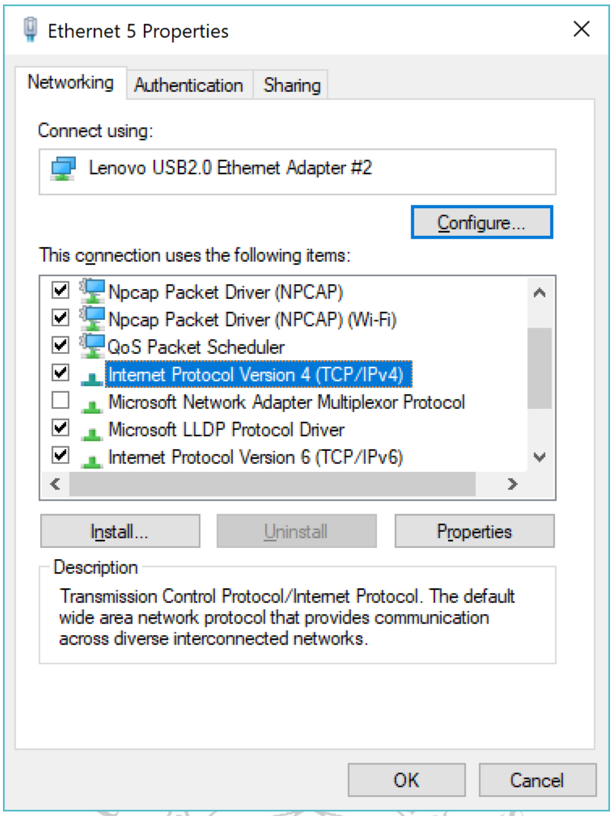


Figure 3.4: Change adapter options

7- Right click and choose properties from the menu. properties, then click on Internet Protocol Version 4:



Ethernet 5 Properties

Connect using:

Lenovo USB2.0 Ethernet Adapter #2

This connection uses the following items:

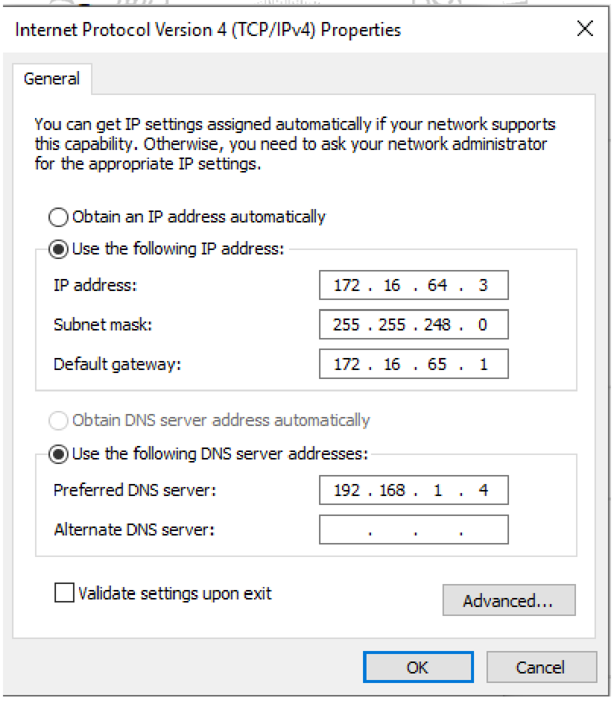
- ☑ Npcap Packet Driver (NPCAP)
- ☑ Npcap Packet Driver (NPCAP) (Wi-Fi)
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- □ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☑ Internet Protocol Version 6 (TCP/IPv6)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Figure 3.5: Ethernet Properties window

8- Then add the static IP, Subnet and gateway according to your network address available.



Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 172 . 16 . 64 . 3

Subnet mask: 255 . 255 . 248 . 0

Default gateway: 172 . 16 . 65 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 192 . 168 . 1 . 4

Alternate DNS server: . . .

□ Validate settings upon exit

Figure 3.6: Internet Protocol Version 4 (TCP/IPv4) Properties

9- Repeat steps 1 to 8 for the other PC.

10- Configure IP addresses on the link connecting both the routers.

11- Configure static routes for routing.

12- Ping the other computer from your PC to verify that connectivity has been established.

13- Turn on network discovery and file sharing options for both the systems.

14- Right click the folder which you want to share. Go to Properties -> Sharing and click on share… . Select **Everyone** from drop-down menu. You can also select **Advanced Sharing** option for further settings.

EXERCISE

1. Connect another PC within your own network (In LAN). Share a folder named "***CCN_TE***" containing the softcopy of lab manual. Access the folder from another PC. Attach screenshots of both PCs showing the folder.

2. Repeat 1. for system in another network.

Attach screenshots here

Lab Session 04

OBJECT

Configure static routes on Cisco routers

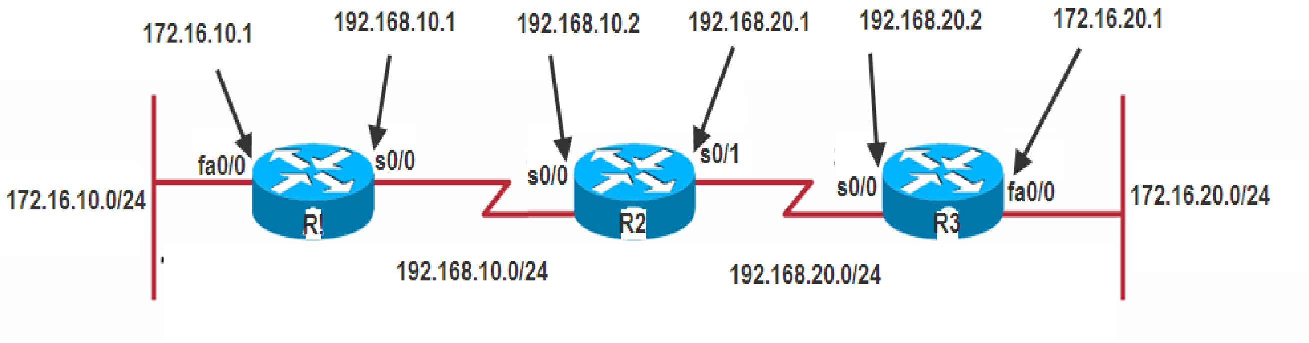


Figure 4.1: Scenario for static routes

THEORY

Routed & Routing Protocols

- A **Routed Protocol** is a protocol by which data can be routed. Routed protocols are IP, AppleTalk, and IPX. In this kind of protocols we require an addressing scheme and sub netting. Addressing scheme will be used to determine the network to which a host belongs and to identifying that host on that particular network. All hosts on an internetwork use the services of a routed protocol.
- A **Routing Protocol** is different and is only used between routers. It makes possible for routers to build and maintain routing tables. There are three classes of routing protocols-
  1) Distance Vector,
  2) Link State,
  3) Hybrid

Static & Dynamic Routing

The simplest method to route packets on a network is static routes. Although dynamic routing protocols are flexible and adjust to network changes, they do have associated network traffic which competes for network bandwidth with the user data traffic.

Configuring Static Routes

Static routes specify a fixed route for a certain destination network. They need to be configured on any router that needs to reach a network that it is not directly connected to. The IOS command used to configure static routes is `ip route`. The syntax is:

`ip route` *destination-address subnet-mask* **{***ip-address* | *outgoing-interface***}** [*distance*] [**tag** *tag*] [*permanent*] where:

- *destination-address* is the destination address prefix for the network that we would like the router to reach
- *subnet-mask* is the subnet mask to be used on the address prefix to match for destination addresses. Multiple networks may be combined such that the destination-address and subnet-mask combination matches all hosts on those networks.
- *ip-address* specifies what ip address to forward a packet to if an IP packet arrives with a destination address that matches the destination-address subnet-mask pair specified in this command.
- Alternatively *outgoing-interface* specifies which interface the packet should be sent out of. Adding a static route to an Ethernet or other broadcast interface (for example, `ip route 0.0.0.0 0.0.0.0 Ethernet 1/2`) will cause the route to be inserted into the routing table only when the interface is up. This configuration is not generally recommended. When the next hop of a static route points to an interface, the router considers each of the hosts within the range of the route to be directly connected through that interface, and therefore it will send ARP requests to any destination addresses that route through the static route.
- *distance* is the optional administrative distance value for the route. If unspecified the default value is 1.
- *tag* value can be used as a "match" value for controlling redistribution via route maps.
- *permanenet* specifies that the route will not be removed even if the interface shuts down.

DTE/DCE

DCE and DTE are the interfaces. The DCE-DTE connection between routers is referred to as a null serial cable DCE (data communication equipment) and DTE (Data terminal equipment). DCE is located at the service provider end while the DTE is attached device.

The services that are given to the DTE is often accessed via modems or channel service unit/data service unit (CSU/DSU). DCE provides clocking and DTE receives the clock

PROCEDURE

1. Connect the network as shown in the network diagram.

2. Configure appropriate ip addresses and clock rates (if needed) on the router interfaces as specified in the network diagram.

3. For R1, enter the following static routes

   ```
   ip route 172.16.20.0 255.255.255.0 192.168.10.2
   ip route 192.168.20.0 255.255.255.0 192.168.10.2
   ```

4. On R2 enter:

   ```
   ip route 172.16.10.0 255.255.255.0 192.168.10.1
   ip route 172.16.20.0 255.255.255.0 192.168.20.2
   ```

5. On R3 enter:

```
ip route 172.16.10.0 255.255.255.0 192.168.20.1
ip route 192.168.10.0 255.255.255.0 192.168.20.1
```

6. After that verify the static routes by entering the following commands in the privilege mode:

   router# sh ip route

EXERCISES

1. Run the command show IP route and write its output.

____

____

____

____

2. What is the default administrative distance of static route? Write the IP route command to modify the same.

____

____

____

____

3. Create a loop back interface on R3 and assign an IP address 10.1.0.1 /16 to it. Now add static routes to each of the other routers to reach this interface. Verify your work by pinging the newly created interface from routers R1 and R2 respectively.

____

____

____

____

____

Lab Session 05

OBJECT

Configure RIP (Routing Information Protocol) and RIP version 2

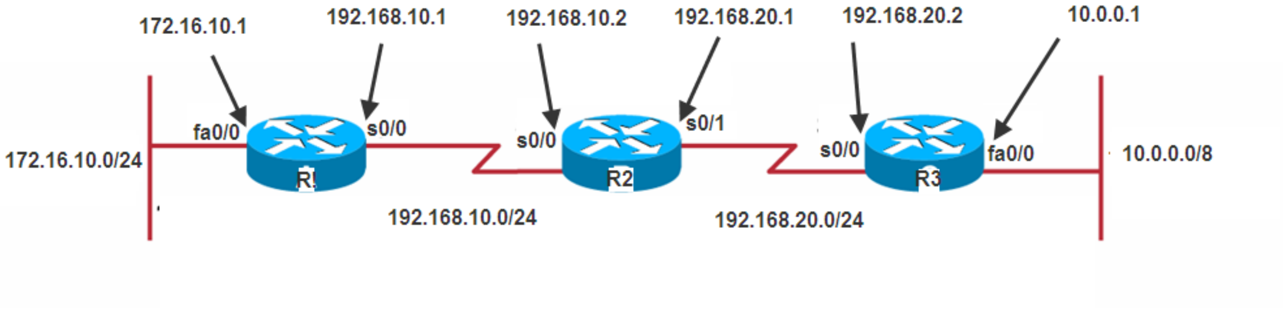


Figure 5.1: Scenario for RIP

THEORY

Distance Vector Routing Protocols

- Broadcast their entire routing table to each neighbor router at predetermined intervals
- The actual interval depends on the distance-vector routing protocol in use
- Varies between 30 and 90 seconds
- Sometimes referred to as *routing by rumor*
- Suffer from slow time to *convergence*
- *Convergence* is an state where all routers on the internetwork share a common view of the internetwork routes

Routing Information Protocol (RIP)

A distance-vector protocol, RIP was designed to work with small to medium-sized networks. RIP is an Interior Gateway Protocol (IGP), meaning it is used within an autonomous system. An autonomous system is a collection of networks under a single administration, sharing a common routing strategy.

RIP is easy to implement, compared to newer IGPs, and has been implemented in networks around the world. Advantage of using RIP, especially in small networks, is that there is very little overhead, in terms of bandwidth used and configuration and management time.

RIP Timers

RIP uses timers both to regulate its performance and to help prevent routing loops. All routers that use RIP send an update message to all of their neighbors approximately every 30 seconds; this process is termed *advertising*. The Cisco implementation sends updates every 30 seconds minus up to 15 percent, or 4.5 seconds.

If a neighbor has not responded in 180 seconds, it is assumed that the neighboring router is unavailable or the network connecting it to the router has become unusable. When the neighbor has not responded for 180 seconds, the route is marked invalid; 180 seconds is long enough that a route won't be invalidated by a single missed update message. The neighbor is shown to be unreachable by sending a normal update message with a metric of "infinity;" in the case of RIP, this number is 16. If an advertisement is received from a neighbor with a metric of infinity, then the route is placed into hold-down state, advertised with a distance of 16, and kept in the routing table. No updates from other neighbors for the same route are accepted while the route is in hold-down state. If other neighbors are still advertising the same route when the hold-down timer expires, then their updates will then be accepted. The route will be advertised with infinity metric for a period of time after the hold-down state if no alternate paths are found.

The actual timers used to accomplish the above tasks are a *routing-update timer*, a *route-invalid timer*, a *route-hold-down timer*, and a *route-flush* timer. The RIP routing-update timer is generally set to 30 seconds, ensuring that each router will send a complete copy of its routing table to all neighbors every 30 seconds. The route-invalid timer determines how much time must expire without a router having heard about a particular route before that route is considered invalid. When a route is marked invalid or put in hold-down state, neighbors are notified of this fact. This notification must occur prior to expiration of the route-flush timer. When the route flush-timer expires, the route is removed from the routing table. Typical initial values for these timers are 180 seconds for the route-invalid and route-hold-down timers and 240 seconds for the route-flush timer. The values for each of these timers can be adjusted with the `timers basic` router configuration command.

Several Stability Features

To adjust for rapid network-topology changes, RIP specifies numerous stability features that are common to many routing protocols. RIP implements split horizon with poison-reverse and hold-down mechanisms to prevent incorrect routing information from being propagated. Split horizon prevents incorrect messages from being propagated by not advertising routes over an interface that the router is using to reach the route. Implementing split horizon helps avoid routing loops. Poison reverse operates by advertising routes that are unreachable with a metric of infinity back to the original source of the route. Hold-down is a method of marking routes invalid (expired). As discussed above, no updates from other neighbors for the same route are accepted while the route is in hold-down state.

Triggered updates are also an included convergence and stability feature. Updates are triggered whenever a metric for a route changes. Triggered updates may also contain only information regarding routes that have changed, unlike scheduled updates.

RIP version 2

RIPv2 is almost the same as the RIP version 1. RIPv2 also sends its complete routing table to its active interfaces at periodic time intervals. The timers, loop avoidance schemes and administrative distance are the same as Rip version 1. But RIPv2 is considered classless routing protocol because it also sends subnet information's with each router. It also allows authentication using MD5 encryption scheme. And it also supports dis-contiguous networks. Configuring RIP version 2 on a router is very simple; it just requires one additional command.

PROCEDURE

Configuring RIP

1. Cable up the network as shown in the diagram.

2. Assign the IP address as shown in the diagram to the appropriate interfaces. For the serial links, has been used to indicate a DCE port.

3. Issue RIP routing commands on all the routers starting from the global config mode.

4. On R1:

   ```
   router rip
   network 172.16.10.0
   network 192.168.10.0
   ```

   On R2

   ```
   router rip
   network 192.168.10.0
   network 192.168.20.0
   ```

   On R 3

   ```
   router rip
     network 10.0.0.0
     network 192.168.20.0
   ```

5. To verify the working of RIP ping one host, say H2, on LAN connected to R3 from the host, say H1, on LAN connected to R1. Also run some other debugging command to explore more.

Configuring RIP version 2

1. Issue the following commands on R1.

   ```
   router rip
   version 2
   network 172.16.10.0
   network 192.168.10.0
   ```

2. Repeat the same for R2 and R3.
3. Verify and debug, as you did earlier for RIP.

EXERCISES

1. Configure RIP on all three routers, note down routing table of router R1, and run command Debug ip rip to note the address on which updates are sent.

____

____

____

____

2. Write commands to modify the default update and hold-down timers.

____

____

____

____

3. Repeat exercise #1 for RIPv2 and note down the multicast address on which RIPv2 forwards the updates.

____

____

____

____

4. Write down the source IP address for the ping packets when you ping H1 from R1.

____

____

____

____

5. While working on R1, how could you check if H1 can reach the loopback interface? In other words, how can you verify if a ping from H1 to loopback of R1 is successful?

____

____

____

____

Lab Session 06

OBJECT

Configuring OSPF (Open Shortest Path First) Single Area

THEORY

Open Shortest Path First (OSPF) was developed by the Internet Engineering Task Force (IETF) as a replacement for the problematic RIP and is now the IETF-recommended Interior Gateway Protocol (IGP). OSPF is a link state protocol that, as the name implies, uses Dijkstra's Shortest Path First (SPF) algorithm. It is an open standards protocol—that is, it isn't proprietary to any vendor or organization. Link-state routing protocols perform the following functions:

- Respond quickly to network changes
- Send triggered updates only when a network change has occurred
- Send periodic updates known as *link-state refreshes*
- Use a *hello mechanism* to determine the reachability of neighbors
    - Each router keeps track of the state or condition of its directly connected neighbors by multicasting hello packets
- Each router also keeps track of all the routers in its network or area of the network by using *link-state advertisements (LSAs).*

Like all link state protocols, OSPF's major advantages over distance vector protocols are fast convergence, support for much larger internetworks, and less susceptibility to bad routing information. Other features of OSPF are:

- The use of areas, which reduces the protocol's impact on CPU and memory, contains the flow of routing protocol traffic, and makes possible the construction of hierarchical internetwork topologies
- Fully classless behavior, eliminating such class-full problems as dis-contiguous subnets. Support of classless route table lookups, VLSM, and super-netting for efficient address management
- A dimensionless, arbitrary metric
- Equal-cost load balancing for more efficient use of multiple paths·
- Support of authentication for more secure routing
- The use of route tagging for the tracking of external routes

Characteristics of OSPF

Characteristic	**OSPF**
VLSM support	Yes
Manual summarization	Yes

Type of protocol	Link state
Classless support	Yes
Auto-summarization	No
Dis-contiguous support	Yes
Route propagation	Multicast on change
Hop count limit	None
Convergence	Fast
Peer authentication	Yes
Hierarchical network Updates/ Route computation	Event triggered/ Dijkstra

DR and BDR

DR (Designated Routers)

DR has the following duties:

- To represent the multi-access network and its attached routers to the rest of the internetwork
- To manage the flooding process on the multi-access network.
- The concept behind the DR is that the network itself is considered a "pseudo node," or a virtual router. Each router on the network forms an adjacency with the DR which represents the pseudo-node. Only the DR will send LSAs to the rest of the internetwork.

Note: router might be a DR on one of its attached multi-access networks, and it might not be the DR on another of its attached multi-access networks. In other words, the DR is a property of a router's interface, not the entire router.

BDR (Backup Designated Router):

A *Backup Designated Router (BDR)* is a hot standby for the DR on multi-access links. The BDR receives all routing updates from OSPF adjacent routers but doesn't flood LSA updates.

Note: if the router interface priority value is set to zero then that router won't participate in the DR or BDR elections on that interface.

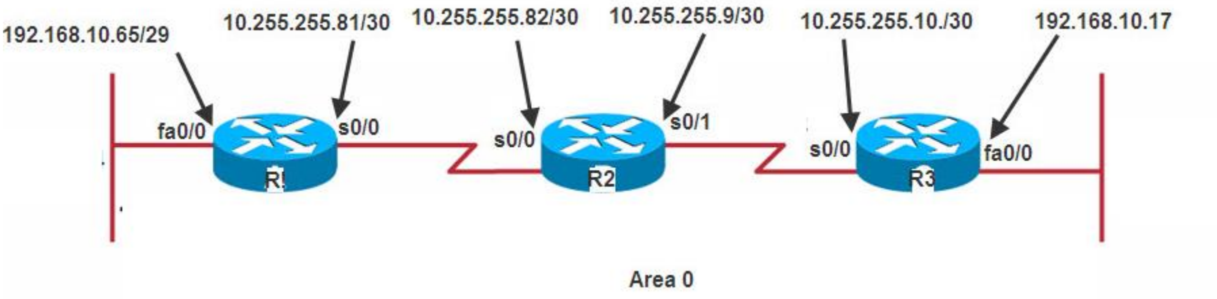


Fig 6.1: Scenario for OSPF implementation

After assigning ip addresses to interfaces of the routers the following IP Routing commands of OSPF on each other will be given as below.

```
Router A:

Router_A#config t
Router_A(config)#router ospf 1
Router_A(config-router)#network 192.168.10.64 0.0.0.7 area 0
Router_A(config-router)#network 10.255.255.80 0.0.0.3 area 0
```

The Router_A is using a /29 or 255.255.255.248 mask on the fa0/0 interface. This is a block size of 8, which is a wildcard of 7. The s0/0 interface is a mask of 255.255.255.252 block size of 4, with a wildcard of 3. Similarly the other subnet mask, and wildcard can be determined by looking at the IP address of an interface.

```
Router B:

Router_B#config t
Router_B(config)#router ospf 1
Router_B(config-router)#network 10.255.255.80 0.0.0.3 area 0
Router_B(config-router)#network 10.255.255.8 0.0.0.3 area 0

Router C:
Router_C#config t
Router_C(config)#router ospf 1
Router_C(config-router)#network 192.168.10.16 0.0.0.7 area 0
Router_C(config-router)#network 10.255.255.8 0.0.0.3 area 0
```

EXERCISES

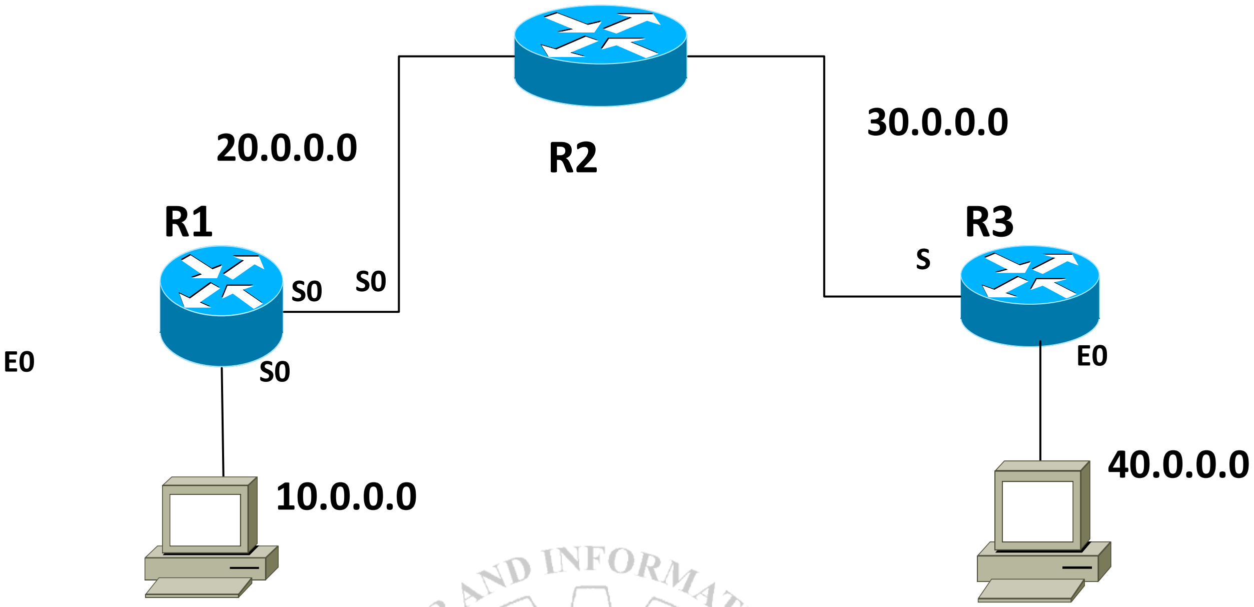


Fig 6.2: Scenario for exercise problems

Simulate the network shown above on packet tracer. Assign appropriate IP addresses on the interfaces and configure OSPF on the routers. Write down the configuration commands entered on all three routers for configuration of OSPF.

1. Router 1:

____

____

____

____

____

____

____

____

____

2. Router 2:

____

____

3. Router 3:

Lab Session 07

OBJECT

Practice how to connect two routers (Branch office and Head office) with the help of PPP

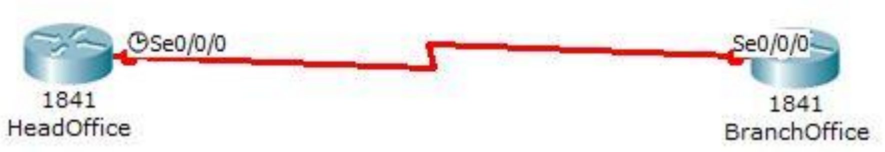
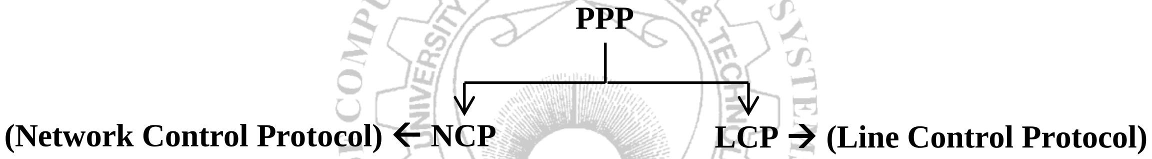


Fig 7.1: Scenario for exercise problems

THEORY

PPP (Point-To-Point Protocol)

Short for **P***oint-to-***P***oint* **P***rotocol*, PPP is a method of connecting a computer to the Internet. PPP is more stable than the older SLIP protocol and provides error checking features. Working in the data link layer of the OSI model, PPP sends the computer's TCP/IP packets to a server that puts them onto the Internet.


PPP
(Network Control Protocol) ← NCP
LCP → (Line Control Protocol)

NCP

A Network Control Protocol is a protocol that runs atop the Point-to-Point Protocol (PPP) and that is used to negotiate options for a network layer protocol running atop PPP. Network Control Protocols include the Internet Protocol Control Protocol for the Internet Protocol, the Internetwork Packet Exchange Control Protocol for the Internet Packet Exchange protocol, and the AppleTalk Control Protocol for AppleTalk. This protocol operates on the data link layer.

LCP

Short for Link Control Protocol, a protocol that is part of the PPP. In PPP communications, both the sending and receiving devices send out LCP packets to determine specific information that will be required for the data transmission. The LCP checks the identity of the linked device and either accepts or rejects the peer device, determines the acceptable packet size for transmission, searches for errors in configuration and can terminate the link if the parameters are not satisfied. Data cannot be transmitted over the network until the LCP packet determines that the link is acceptable.

Authentications methods of PPP

PAP and CHAP are two methods that PPP uses for authentication.

PAP (Password Authentication Protocol)

Short for Password Authentication Protocol, the most basic form of authentication, in which a user's name and password are transmitted over a network and compared to a table of name-password pairs. Typically, the passwords stored in the table are encrypted. The Basic Authentication feature built into the HTTP protocol uses PAP. The main weakness of PAP is that both the username and password are transmitted "in the clear" -- that is, in an unencrypted form. It’s a two way hand shake method.

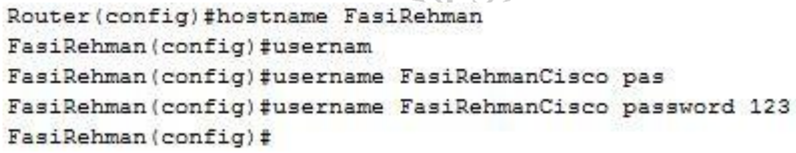
CHAP (Challenge Handshake Authentication Protocol)

Short for Challenge Handshake Authentication Protocol, CHAP is a type of authentication in which the authentication agent (typically a network server) sends the client program a random value that is used only once and an ID value. Both the sender and peer share a predefined secret. The peer concatenates the random value (or nonce), the ID and the secret and calculates a one-way hash using MD5. The hash value is sent to the authenticator, which in turn builds that same string on its side, calculates the MD5 sum itself and compares the result with the value received from the peer. If the values match, the peer is authenticated .By transmitting only the hash, the secret can't be reverse-engineered. The ID value is increased with each CHAP dialogue to protect against replay attacks. It’s a three way hand shake method.

PROCEDURE

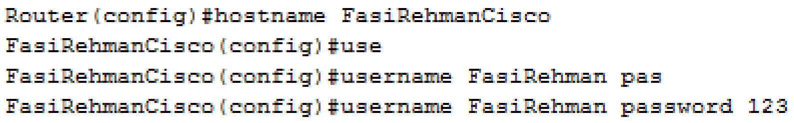
1. Change Hostname and assign Username and password on both routers. Assign the username of Branch Office Router in Head Office Router and Head Office Router username initialized on Branch Office Router but Password must be same on both Routers.

   HeadOffice Router:



```
Router(config)#hostname FasiRehman
FasiRehman(config)#usernam
FasiRehman(config)#username FasiRehmanCisco pas
FasiRehman(config)#username FasiRehmanCisco password 123
FasiRehman(config)#
```

   BranchOffice Router:

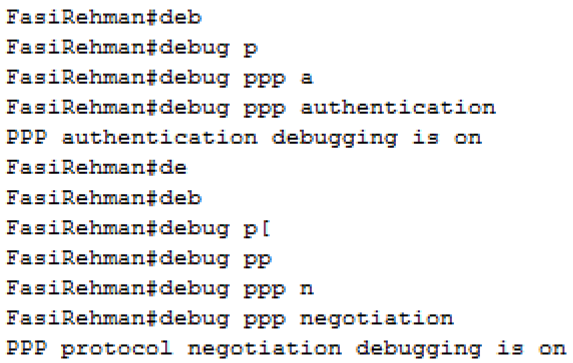


```
Router(config)#hostname FasiRehmanCisco
FasiRehmanCisco(config)#use
FasiRehmanCisco(config)#username FasiRehman pas
FasiRehmanCisco(config)#username FasiRehman password 123
```

2. Issue PPP debugging commands in privileged mode of both routers.

   - **debug ppp authentication:** The most common reasons for failed dial backup calls are incorrect dial strings and PPP authentication problems. You can easily diagnose both of these problems with this command.

- **debug ppp negotiation:** Displays PPP packets related to the negotiation of the PPP link.



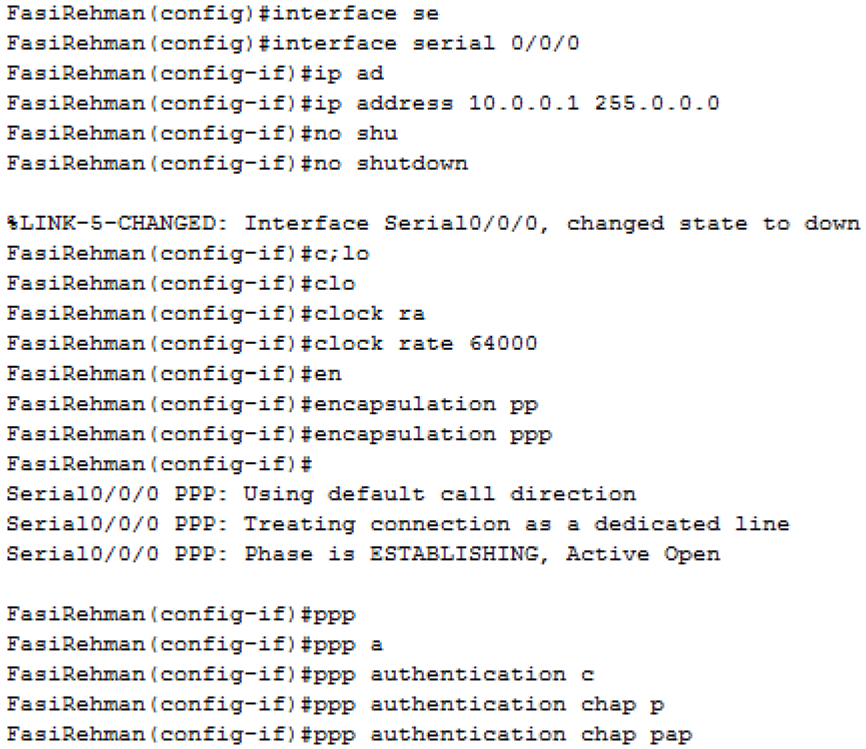
```
FasiRehman#deb
FasiRehman#debug p
FasiRehman#debug ppp a
FasiRehman#debug ppp authentication
PPP authentication debugging is on
FasiRehman#de
FasiRehman#deb
FasiRehman#debug p[
FasiRehman#debug pp
FasiRehman#debug ppp n
FasiRehman#debug ppp negotiation
PPP protocol negotiation debugging is on
```

3. Assingning IP addresses on serial interfaces and enabling PPP on both routers

   - **encapsulation ppp:** Change encapsulation from default HDLC to PPP

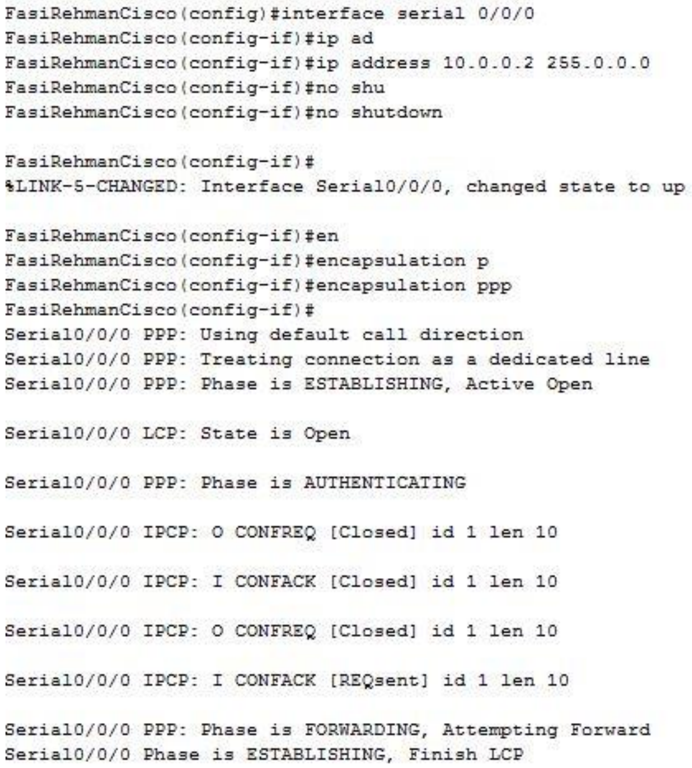
   - **ppp authentication chap pap:** Define that the Link will use PAP authentication, but will try CHAP if PAP fails or is rejected by other side.

HeadOffice Router:

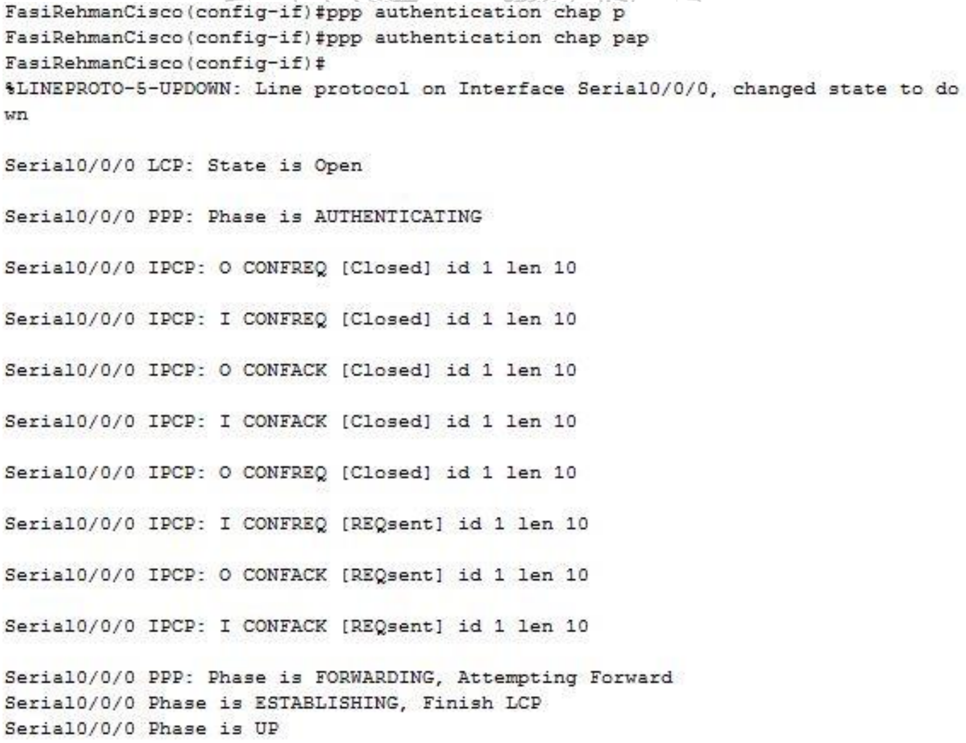


```
FasiRehman(config)#interface se
FasiRehman(config)#interface serial 0/0/0
FasiRehman(config-if)#ip ad
FasiRehman(config-if)#ip address 10.0.0.1 255.0.0.0
FasiRehman(config-if)#no shu
FasiRehman(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
FasiRehman(config-if)#c;lo
FasiRehman(config-if)#clo
FasiRehman(config-if)#clock ra
FasiRehman(config-if)#clock rate 64000
FasiRehman(config-if)#en
FasiRehman(config-if)#encapsulation pp
FasiRehman(config-if)#encapsulation ppp
FasiRehman(config-if)#
Serial0/0/0 PPP: Using default call direction
Serial0/0/0 PPP: Treating connection as a dedicated line
Serial0/0/0 PPP: Phase is ESTABLISHING, Active Open

FasiRehman(config-if)#ppp
FasiRehman(config-if)#ppp a
FasiRehman(config-if)#ppp authentication c
FasiRehman(config-if)#ppp authentication chap p
FasiRehman(config-if)#ppp authentication chap pap
```

BranchOffice Router:



```
FasiRehmanCisco(config)#interface serial 0/0/0
FasiRehmanCisco(config-if)#ip ad
FasiRehmanCisco(config-if)#ip address 10.0.0.2 255.0.0.0
FasiRehmanCisco(config-if)#no shu
FasiRehmanCisco(config-if)#no shutdown

FasiRehmanCisco(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

FasiRehmanCisco(config-if)#en
FasiRehmanCisco(config-if)#encapsulation p
FasiRehmanCisco(config-if)#encapsulation ppp
FasiRehmanCisco(config-if)#
Serial0/0/0 PPP: Using default call direction
Serial0/0/0 PPP: Treating connection as a dedicated line
Serial0/0/0 PPP: Phase is ESTABLISHING, Active Open

Serial0/0/0 LCP: State is Open

Serial0/0/0 PPP: Phase is AUTHENTICATING

Serial0/0/0 IPCP: O CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFACK [Closed] id 1 len 10

Serial0/0/0 IPCP: O CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFACK [REQsent] id 1 len 10

Serial0/0/0 PPP: Phase is FORWARDING, Attempting Forward
Serial0/0/0 Phase is ESTABLISHING, Finish LCP
```



```
FasiRehmanCisco(config-if)#ppp authentication chap p
FasiRehmanCisco(config-if)#ppp authentication chap pap
FasiRehmanCisco(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to do
wn

Serial0/0/0 LCP: State is Open

Serial0/0/0 PPP: Phase is AUTHENTICATING

Serial0/0/0 IPCP: O CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: O CONFACK [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFACK [Closed] id 1 len 10

Serial0/0/0 IPCP: O CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFREQ [REQsent] id 1 len 10

Serial0/0/0 IPCP: O CONFACK [REQsent] id 1 len 10

Serial0/0/0 IPCP: I CONFACK [REQsent] id 1 len 10

Serial0/0/0 PPP: Phase is FORWARDING, Attempting Forward
Serial0/0/0 Phase is ESTABLISHING, Finish LCP
Serial0/0/0 Phase is UP
```

EXERCISES

1. Run the command show ppp authentication and write its output.

2. Write down the difference between chap and pap

3. Write down the command to use chap and pap as a fall back method to one another

Lab Session 08

OBJECT

Configure Access Lists

THEORY

An access list is essentially a list of conditions that categorize packets. One of the most common and easiest to understand uses of access lists is filtering unwanted packets when implementing security policies. Access lists can even be used in situations that don't necessarily involve blocking packets.

There are a few important rules that a packet follows when it's being compared with an access list:

Rule#1
It's always compared with each line of the access list in sequential order—that is, it'll always start with the first line of the access list, then go to line 2, then line 3, and so on.

Rule#2
It's compared with lines of the access list only until a match is made. Once the packet matches the condition on a line of the access list, the packet is acted upon and no further comparisons take place.

Rule#3
There is an implicit "deny" at the end of each access list—this means that if a packet doesn't match the condition on any of the lines in the access list, the packet will be discarded. Each of these rules has some powerful implications when filtering IP packets with access lists, so keep in mind that creating effective access lists truly takes some practice.

There are two main types of access lists:

1. Standard access lists
2. Extended access lists

Standard access lists

These use only the source IP address in an IP packet as the condition test. All decisions are made based on the source IP address. This means that standard access lists basically permit or deny an entire suite of protocols. They don't distinguish between any of the many types of IP traffic such as web, Telnet, UDP, and so on.

Its command syntax is

```
access-list <number> {permit| deny} <destination> [log]
```

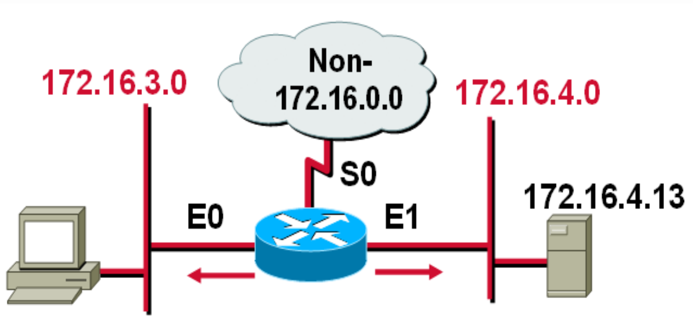


Fig 8.1: Standard Access list to allow my network

Commands on router will be

```
R1(config)#aaccess-list 1 permit 172.16.0.0 0.0.255.255
R1(config)#interface ethernet 0
R1(config)#ip access-group 1 out
R1(config)#interface ethernet 1
R1(config)#ip access-group 1 out
```

The above commands will permit the network 172.16.0.0 only and will block other network through the router on its ethernet interfaces in its outside directions

Extended access lists

Extended access lists can evaluate many of the other fields in the layer 3 and layer 4 headers of an IP packet. They can evaluate source and destination IP addresses, the protocol field in the Network layer header, and the port number at the Transport layer header. This gives extended access lists the ability to make much more granular decisions when controlling traffic.

Its command syntax is

```
access-list <number> {permit| deny}
<protocol><source>[<ports>]<destination>[ports][<options>]
```

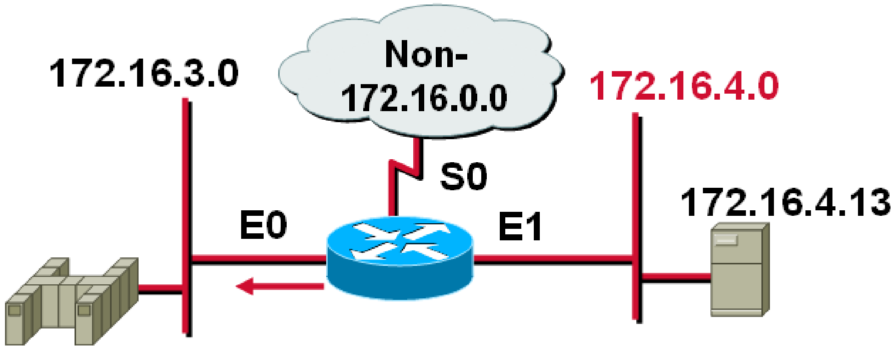


Fig 8.2: Extended access list

Commands on the router will be:

```
access-list 101 deny tcp 172.16.4.0 0.0.0.255 any eq 23
access-list 101 permit ip any any
interface ehternet 0
ip access-group 101 out
```

The above commands will Deny only the Telnet from subnet 172.16.40.0 out of E0 and will permit all other traffic.

EXERCISE

1. Give commands to enable logging for the given access list and to show the entries that have been blocked

____

____

____

____

____

____

____

Lab Session 09

OBJECT

Explore basic LAN Switch Operation and Loop Avoidance with Spanning Tree.

THEORY

LAN switch performs 3 operations

- Address learning
- Forward filter decision
- Loop avoidance

In this session, we will explore how an Ethernet switch learns addresses of the attached hosts.

Address learning

A new switch has empty MAC address table. As each frame transits switch, it learns source MAC address against the source port. As the switch does not know to which port the destination is attached, it initially transmits the frame to all ports. This process is called flooding. As the responses are received, the MAC address table is further populated.

PROCEDURE

Consider the following scenario

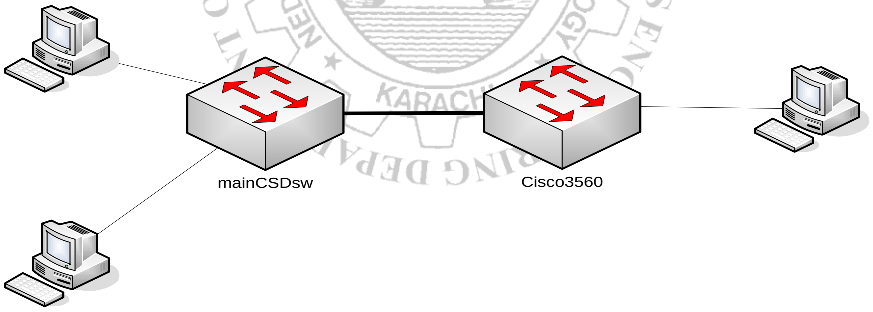


Fig 9.1: Scenario for LAN switch operation

Initially the MAC database of Cisco3560 will be

```
Switch#sh mac-address-table


          Mac Address Table
        -------------------------------------------

        Vlan    Mac Address       Type        Ports
        ----    -----------       --------    -----

           1    0006.2a75.100c    DYNAMIC     Fa0/1
Switch#
```

And that of mainCISDsw is;

```
mainCISDsw#sh mac-address-table

        Mac Address Table
      -------------------------------------------

      Vlan    Mac Address       Type        Ports
      ----    -----------       --------    -----

         1    0060.471b.ae01    DYNAMIC     Eth0/1

mainCISDsw#
```

Now as any of the computers generates ping for any of the remaining computers, the MAC address table will grow

```
Switch#sh mac-address-table

      Mac Address Table
      -------------------------------------------

      Vlan    Mac Address       Type        Ports
      ----    -----------       --------    -----

         1    0006.2a75.100c    DYNAMIC     Fa0/1
         1    0040.0ba5.183a    DYNAMIC     Fa0/1
         1    00e0.f7a4.475c    DYNAMIC     Fa0/2
Switch#
```

Also for mainCSDsw

```
mainCISDsw#sh mac-address-table

       Mac Address Table
      -------------------------------------------

      Vlan    Mac Address       Type        Ports
      ----    -----------       --------    -----

         1    0040.0ba5.183a    DYNAMIC     Eth1/1
         1    0060.471b.ae01    DYNAMIC     Eth0/1
         1    00e0.f7a4.475c    DYNAMIC     Eth0/1
mainCISDsw#
```

Spanning Tree Protocol (STP)

The **Spanning Tree Protocol (STP)** is a link layer network protocol that ensures a loop-free topology for any switched LAN. Thus, the basic function of STP is to prevent switching loops and ensuing broadcast radiation.

In the OSI model for computer networking, STP falls under the OSI layer-2. It is standardized as 802.1D. As the name suggests, it creates a spanning tree within a mesh network of connected layer-2 switches (typically Ethernet switches), and disables those links that are not part of the spanning tree, leaving a single active path between any two network nodes.

Spanning tree allows a network design to include spare (redundant) links to provide automatic backup paths if an active link fails, without the danger of switch loops, or the need for manual enabling/disabling of these backup links. Switch loops must be avoided because they result in flooding the local network.

STP is based on an algorithm invented by Radia Perlman while working for Digital Equipment Corporation.

Protocol Operation

The collection of switches in a LAN can be considered a graph whose nodes are the bridges and the LAN segments (or cables), and whose edges are the interfaces connecting the bridges to the segments. To break loops in the LAN while maintaining access to all LAN segments, the bridges collectively compute a spanning tree. The spanning tree is not necessarily a minimum cost spanning tree. A network administrator can reduce the cost of a spanning tree, if necessary, by altering some of the configuration parameters in such a way as to affect the choice of the root of the spanning tree.

The spanning tree that the bridges compute using the Spanning Tree Protocol can be determined using the following rules.

Select a root bridge. The *root bridge* of the spanning tree is the bridge with the smallest (lowest) bridge ID. Each bridge has a unique identifier (ID) and a configurable priority number; the bridge ID contains both numbers. To compare two bridge IDs, the priority is compared first. If two bridges have equal priority, then the MAC addresses are compared. For example, if switches A (MAC=0200.0000.1111) and B (MAC=0200.0000.2222) both have a priority of 10, then switch A will be selected as the root bridge. If the network administrators would like switch B to become the root bridge, they must set its priority to be less than 10.

Determine the least cost paths to the root bridge. The computed spanning tree has the property that messages from any connected device to the root bridge traverse a least cost path, i.e., a path from the device to the root that has minimum cost among all paths from the device to the root. The cost of traversing a path is the sum of the costs of the segments on the path. Different technologies have different default costs for network segments. An administrator can configure the cost of traversing a particular network segment.

The property that messages always traverse least-cost paths to the root is guaranteed by the following two rules.

Least cost path from each bridge. After the root bridge has been chosen, each bridge determines the cost of each possible path from itself to the root. From these, it picks one with the smallest cost (a least-cost path). The port connecting to that path becomes the *root port* (RP) of the bridge.

Least cost path from each network segment. The bridges on a network segment collectively determine which bridge has the least-cost path from the network segment to the root. The port connecting this bridge to the network segment is then the *designated port* (DP) for the segment.

Disable all other root paths. Any active port that is not a root port or a designated port is a *blocked port* (BP).

Bridge Protocol Data Units (BPDUs)

The above rules describe one way of determining what spanning tree will be computed by the algorithm, but the rules as written require knowledge of the entire network. The bridges have to determine the root bridge and compute the port roles (root, designated, or blocked) with only the information that they have. To ensure that each bridge has enough information, the bridges use special data frames called **Bridge Protocol Data Units** (BPDUs) to exchange information about bridge IDs and root path costs.

A bridge sends a BPDU frame using the unique MAC address of the port itself as a source address, and a destination address of the STP multicast address 01:80:C2:00:00:00.

There are three types of BPDUs:

- Configuration BPDU (CBPDU), used for Spanning Tree computation
- Topology Change Notification (TCN) BPDU, used to announce changes in the network topology
- Topology Change Notification Acknowledgment (TCA)

BPDUs are exchanged regularly (every 2 seconds by default) and enable switches to keep track of network changes and to start and stop forwarding at ports as required.

When a device is first attached to a switch port, it will not immediately start to forward data. It will instead go through a number of states while it processes BPDUs and determines the topology of the network. When a host is attached such as a computer, printer or server the port will always go into the forwarding state, albeit after a delay of about 30 seconds while it goes through the listening and learning states (see below). The time spent in the listening and learning states is determined by a value known as the forward delay (default 15 seconds and set by the root bridge). However, if instead another *switch* is connected, the port may remain in blocking mode if it is determined that it would cause a loop in the network. Topology Change Notification (TCN) BPDUs are used to inform other switches of port changes. TCNs are injected into the network by a non-root switch and propagated to the root. Upon receipt of the TCN, the root switch will set a Topology Change flag in its normal BPDUs. This flag is propagated to all other switches to instruct them to rapidly age out their forwarding table entries.

Spanning Tree port states:

- **Blocking** - A port that would cause a switching loop, no user data is sent or received but it may go into forwarding mode if the other links in use were to fail and the spanning tree algorithm determines the port may transition to the forwarding state. BPDU data is still received in blocking state.
- **Listening** - The switch processes BPDUs and awaits possible new information that would cause it to return to the blocking state.

- **Learning** - While the port does not yet forward frames (packets) it does learn source addresses from frames received and adds them to the filtering database (switching database)
- **Forwarding** - A port receiving and sending data, normal operation. STP still monitors incoming BPDUs that would indicate it should return to the blocking state to prevent a loop.
- **Disabled** - Not strictly part of STP, a network administrator can manually disable a port

Now consider the following topology

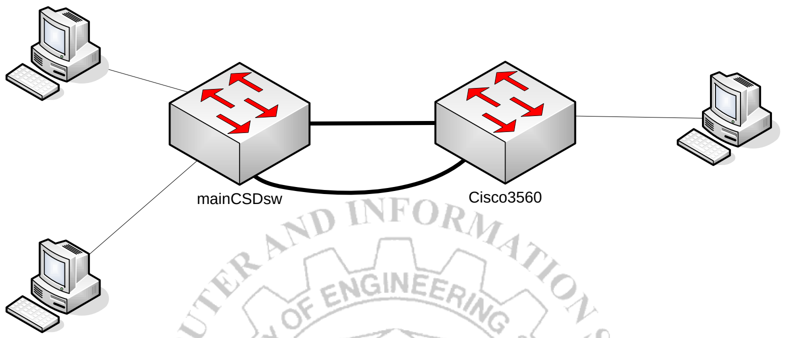


Fig 14.1: Scenario for implementing spanning tree

Here a physical loop can be observed

Now observe the spanning tree calculations for **mainCSDsw** first

```
mainCISDsw#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0010.1100.58CE
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0010.1100.58CE
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- ------------------------------
--
Et3/1            Desg FWD 100       128.4    P2p
Et2/1            Desg FWD 100       128.3    P2p
Et0/1            Desg FWD 100       128.1    P2p
Et1/1            Desg FWD 100       128.2    P2p

For cisco3560 the calculations will be

Switch#sh spanning-tree
VLAN0001
```

```
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0010.1100.58CE
             Cost        100
             Port        1(FastEthernet0/1)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     00E0.B02B.5EA0
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- ------------------------------
--
Fa0/1            Root FWD 100       128.1    P2p
Fa0/3            Altn BLK 100       128.3    P2p
Fa0/2            Desg FWD 19        128.2    P2p
```

Modifying priorities and other parameters

To change default priority one can use the following command.

```
mainCISDsw(config)#spanning-tree vlan 1 priority 36864
```

Now see what happens to the root bridge.

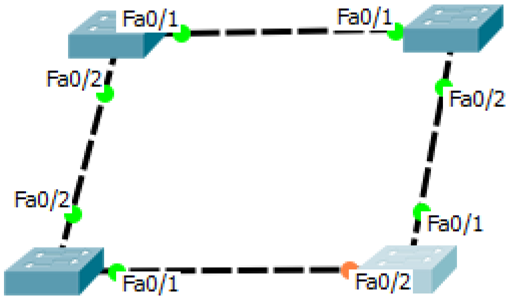
```
mainCISDsw#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     00E0.B02B.5EA0
             Cost        100
             Port        1(Ethernet0/1)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    36865  (priority 36864 sys-id-ext 1)
             Address     0010.1100.58CE
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- ------------------------------
--
Et3/1            Altn BLK 100       128.4    P2p
Et2/1            Desg FWD 100       128.3    P2p
Et0/1            Root FWD 100       128.1    P2p
Et1/1            Desg FWD 100       128.2    P2p
```

Other details on STP can be observed through the following set of commands under spanning tree.

```
Switch#sh spanning-tree ?
  active     Report on active interfaces only
  detail     Detailed information
  interface  Spanning Tree interface status and configuration
  summary    Summary of port states
  vlan       VLAN Switch Spanning Trees
  <cr>
```

EXERCISES

1. Connect the switches as shown below. Identify the root bridge, root ports, designated ports and block port? Toggle the port status of the switch with blocked port and observe the states of the port. Attach screenshots showing different states.



2. Find out what is the default time STP takes to converge from blocking to forwarding state? Justify your answer.

Attach screenshots here

Lab Session 10

OBJECT

Configure Virtual LANs

THEORY

A **virtual LAN**, commonly known as a **VLAN**, is a group of hosts with a common set of requirements that communicate as if they were attached to the same broadcast domain, regardless of their physical location. A VLAN has the same attributes as a physical LAN, but it allows for end stations to be grouped together even if they are not located on the same network switch. Network reconfiguration can be done through software instead of physically relocating devices.

To physically replicate the functions of a VLAN, it would be necessary to install a separate, parallel collection of network cables and switches/hubs which are kept separate from the primary network. However unlike a physically separate network, VLANs must share bandwidth; two separate one-gigabit VLANs using a single one-gigabit interconnection can both suffer reduced throughput and congestion. It virtualizes VLAN behaviors (configuring switch ports, tagging frames when entering VLAN, lookup MAC table to switch/flood frames to trunk links, and untagging when exit from VLAN.)

Implementation

A basic switch not configured for VLANs will either have VLAN functionality disabled, or will have it permanently enabled with what is known as a *default VLAN* which simply contains all ports on the device as members.

Configuration of the first custom VLAN port group usually involves subtracting ports from the default VLAN, such that the first custom group of VLAN ports is actually the second VLAN on the device, apart from the default VLAN. The default VLAN typically has an ID of 1.

If a VLAN port group were to only exist on the one device, all ports that are members of the VLAN group only need to be "untagged". It is only when the port group is to extend to another device that tagging is used. For communications to occur from switch to switch, an uplink port needs to be a tagged member of every VLAN on the switch that uses that uplink port, including the default VLAN.

Some switches either allow or require a name be created for the VLAN, but it is only the VLAN group number that is important from one switch to the next.

Where a VLAN group is to simply pass through an intermediate switch via two pass-through ports, only the two ports need to be a member of the VLAN, and are tagged to pass both the required VLAN and the default VLAN on the intermediate switch.

Management of the switch requires that the management functions be associated with one of the configured VLANs. If the default VLAN were deleted or renumbered without moving the management to a different VLAN first, it is possible to be locked out of the switch configuration, requiring a forced clearing of the device configuration to regain control.

Switches typically have no built-in method to indicate VLAN port members to someone working in a wiring closet. It is necessary for a technician to either have management access to the device to view its configuration, or for VLAN port assignment charts or diagrams to be kept next to the switches in each wiring closet. These charts must be manually updated by the technical staff whenever port membership changes are made to the VLANs.

Remote configuration of VLANs presents several opportunities for a technician to accidentally cut off communications and lock themselves out of the devices they are attempting to configure. Actions such as subdividing the default VLAN by splitting off the switch uplink ports into a separate new VLAN can suddenly cut off all remote communication, requiring the technician to physically visit the device in the distant location to continue the configuration process.

When inside the world of VLANs there are two types of links. These links allow us to connect multiple switches together or just simple network devices e.g PC, that will access the VLAN network. Depending on their configuration, they are called Access Links, or Trunk Links.

Access Links
Access Links are the most common type of links on any VLAN switch. All network hosts connect to the switch's Access Links in order to gain access to the local network. These links are your ordinary ports found on every switch, but configured in a special way, so you are able to plug a computer into them and access your network.

Trunk Link
A Trunk Link, or 'Trunk' is a port configured to carry packets for any VLAN. These type of ports are usually found in connections between switches. These links require the ability to carry packets from all available VLANs because VLANs span over multiple switches.

PROCEDURE

VLAN 1 is the default

```
Switch #sh int vlan 1

Vlan1 is administratively down, line protocol is down
  Hardware is CPU Interface, address is 00e0.b02b.5ea0 (bia 00e0.b02b.5ea0)
  MTU 1500 bytes, BW 100000 Kbit, DLY 1000000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
```

Configuring IP on default VLAN

```
Switch(config)#int vlan 1
Switch(config-if)#ip address 172.16.68.2 255.255.248.0
```

Creating VLANs

```
Switch(config)#int vlan 2
```

Assigning ports to vlans

```
Switch(config)#int fa 0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
```

Configuring trunk link

Consider the following topology

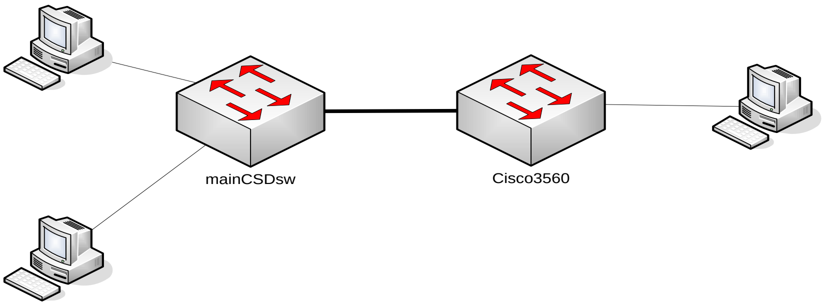


Fig 10.1: Scenario for implementing VLANs

Suppose mainCSDsw has two VLANs configured VLAN1 and VLAN2, whereas cisco3560 has only VLAN1. Now both switched must have at least one common trunk link connecting the two switches, so that the PCs which are in VLAN1 may communicate. Here we have interface fa 0/1 on each switch connected to the other. Hence the configuration would be

```
Switch(config)#int fa 0/1
Switch(config-if)#switchport mode trunk
```

Verification of configurations

```
Switch#show interface switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
```

```
Appliance trust: none

Name: Fa0/2
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
```

A more handy way of verifying VLAN memberships would be

```
mainCISDsw#sh vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------
1    default                          active    Eth2/1, Eth3/1, Eth4/1
2    VLAN0002                         active    Eth1/1
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

EXERCISES

1. What does trunking provide?

2. What type of link is only part of one VLAN and is referred to as the “native VLAN” of the port?

Lab Session 11

OBJECT

Implement Network Address Translation

THEORY

NAT is the one-to-one translation of IP addresses from an inside local IP address to a global IP address that is unique and routable on the Internet.

How NAT Works

Traffic that is sourced on the inside of the network, coming to an interface marked as inside, will have an inside local address as its source IP address and an outside global address as the destination IP address, assuming that no inbound translation of outside addresses is being performed.

The destination would be considered outside local only if bidirectional NAT is being performed. Remember to ask where the source device thinks is the location of the address it is sending to—on the local network or on the other side of the router that is transparently acting as the NAT server. If the source host thinks it's talking to a local device, and the NAT server must step in and provide the smoke and mirrors to make the communication occur with a device that is actually not local, then and only then is the address referred to as an outside *local* address.

When that traffic reaches the NAT process and is switched to the outside network, going out an interface marked as outside, the source IP address will be translated to an inside global address, and the destination IP address will still be known as the outside global address, as it has not changed. Figure 1 shows the inside/outside and local/global relationship.

When traffic is sourced on the outside of the network, coming to an interface marked as outside, the source IP address is known as the outside global address, while the destination IP address is known as the inside global address. When that traffic reaches the NAT process and is switched to the inside network, going out an interface marked as inside, the source IP address will still be known as the outside global address, assuming that it was not translated coming in, and the translated destination IP address will be known as the inside local address. Let's talk about some of the advantages to using NAT.

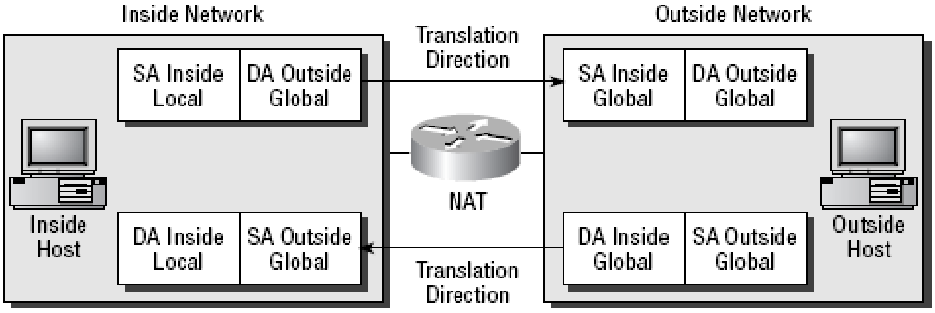


Figure 11.1 NAT inside/outside and local/global relationship

Configuring NAT

We will talk about the commands needed for the following NAT configurations:

- Static NAT
- Dynamic NAT

Configuring Static NAT

Static NAT is used to map a single inside global IP address to a single inside local IP address. Usually the inside local IP address is one from the RFC 1918 address space and the inside global IP address is an Internet-routable address. IP addresses must be assigned to interfaces on the router that will be participating in NAT in order for proper processing of IP traffic on those interfaces. You must be in global configuration mode in order to configure NAT. The command to use is ip nat inside source static *local-ip global-ip*.

The *local-ip* is the local IP address of the host on the inside of the network to translate, and the *global-ip* is the global IP address this inside host will be known as to the outside world. In this example, a host on the inside network needs to access the Internet. Its IP address is 10.1.2.25 and is not routable on the Internet. When the NAT border router receives a packet from 10.1.2.25 destined for the Internet, the router must be configured to translate that IP address to one that is globally routable. In this case it is 200.1.1.25 and the following command is used:

```
Border(config)#ip nat inside source static 10.1.2.25 200.1.1.25
Border(config)#
```

This creates a permanent entry in the NAT table. Now when traffic arrives for IP address 200.1.1.25 from the outside network, it will be translated to 10.1.2.25 and forwarded to the inside network, and vice versa. This allows a device on the inside network using a non-Internet-routable IP address to be accessible from the outside network. You can use this to make your internal DNS, web, etc. servers accessible from the Internet. Optionally, you can configure just a certain port to be translated. Adding a protocol and port numbers to the ip nat inside source static NAT command does this. The following is an example of using an IP address and port combination:

```
Border(config)#ip nat inside source static tcp 10.1.2.25 80
200.1.1.25 80
```

```
Border(config)#ip nat inside source static tcp 10.1.2.24 80
200.1.1.25 81
Border(config)#
```

As you can see, the two port numbers do not need to match on both sides. I have translated requests for port 81 from the outside to port 80 on the inside, which is commonly referred to as port redirection. Now let’s discuss dynamic NAT.

Configuring Dynamic NAT

Dynamic NAT is used to map inside local IP addresses to inside global IP addresses on the fly from a pool of available IP addresses. Again, you must have IP addresses assigned to the interfaces on the router that will be participating in the NAT process in order for IP processing on those interfaces to occur.

The dynamic NAT configuration starts in global configuration mode. In our example network, we will use one interface connected to the inside network (Ethernet0) and one interface connected to the Internet (Serial0). When a host on the inside of the network wants to communicate with a host on the Internet, the NAT border router receives a packet from an interface marked as NAT inside. The outbound interface is marked as NAT outside, and so the router will choose an available IP address from the pool and assign it to the NAT table entry. Once an IP address is allocated, it cannot be allocated to another translation entry until that entry times out or is manually removed.

When traffic goes from inside to outside, NAT translations happen after routing has taken place. Therefore, any access lists or policy routing will have been applied before the NAT translation happens. An access list will need to be created to inform the NAT process what traffic will be translated and what traffic will not. The next step is to configure a pool of IP addresses that will be allocated to outbound sessions. This is done with the ip nat pool command. The syntax of this command is as follows:

`ip nat pool` *`pool-name start-ip end-ip`* `netmask` *`net-mask`*
or
`ip nat pool` *`pool-name start-ip end-ip`* `prefix-length` *`length`*

The *pool-name* is any unique string that identifies this address pool. The *start-ip* and *end-ip* are the starting and ending IP addresses within the pool. The *net-mask* is the network mask in dotted-decimal format that will be used with the addresses in the pool. Optionally, you can use the prefix-length keyword followed by the *length* of the CIDR prefix instead of using a network mask. Finally, you need to tie the access list and pool together with the ip nat inside source command. The following is the syntax of this command:

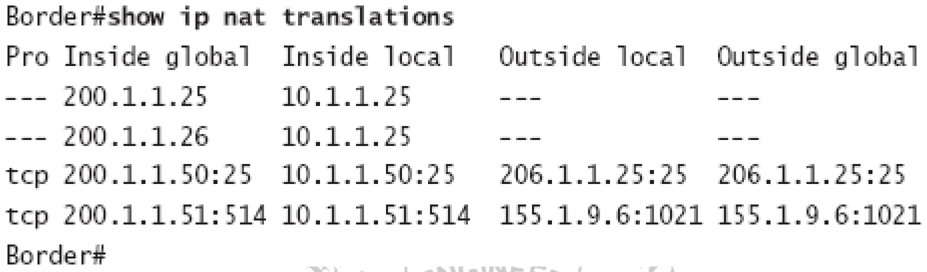
`ip nat inside source list` *`acc-list`* `pool` *`pool-name`*

The *acc-list* is the number or name of the access list you created that specifies the traffic to NAT, and the *pool-name* is the unique string used when you created the pool of IP addresses. The following is an example of configuring dynamic NAT using a pool:

```
Border(config)#interface ethernet0
Border(config-if)#ip nat inside
Border(config-if)#interface serial0
Border(config-if)#ip nat outside
Border(config-if)#exit
Border(config)#access-list 12 permit 10.1.2.0 0.0.0.255
Border(config)#ip nat pool OUTBOUND 200.1.1.2 200.1.1.254 prefix-
length 24
Border(config)#ip nat inside source list 12 pool OUTBOUND
Border(config)#
```

Verifying and Troubleshooting the NAT Configuration

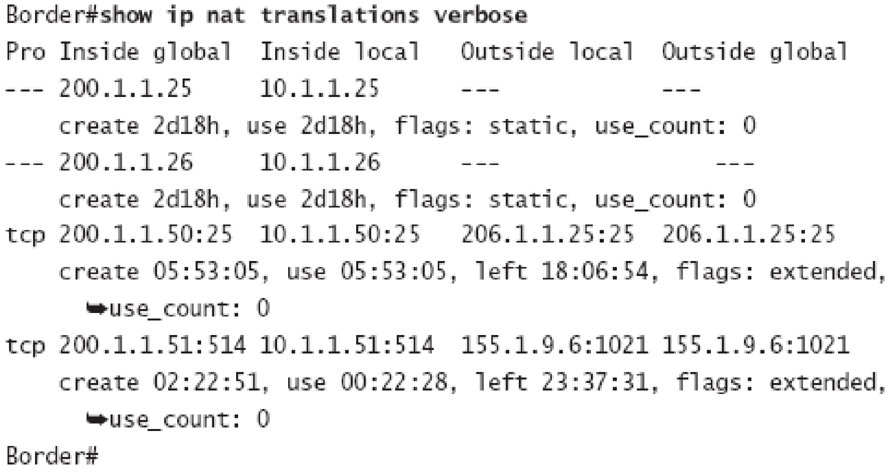
There are two commands used to verify the NAT configuration on a router. The *show ip nat translations* command shows the translations in the NAT table: The following is an example of its output as shown in fig 3.2:



```
Border#show ip nat translations
Pro Inside global  Inside local   Outside local  Outside global
--- 200.1.1.25     10.1.1.25      ---            ---
--- 200.1.1.26     10.1.1.25      ---            ---
tcp 200.1.1.50:25  10.1.1.50:25   206.1.1.25:25  206.1.1.25:25
tcp 200.1.1.51:514 10.1.1.51:514  155.1.9.6:1021 155.1.9.6:1021
Border#
```

Figure 11.2: NAT translation table

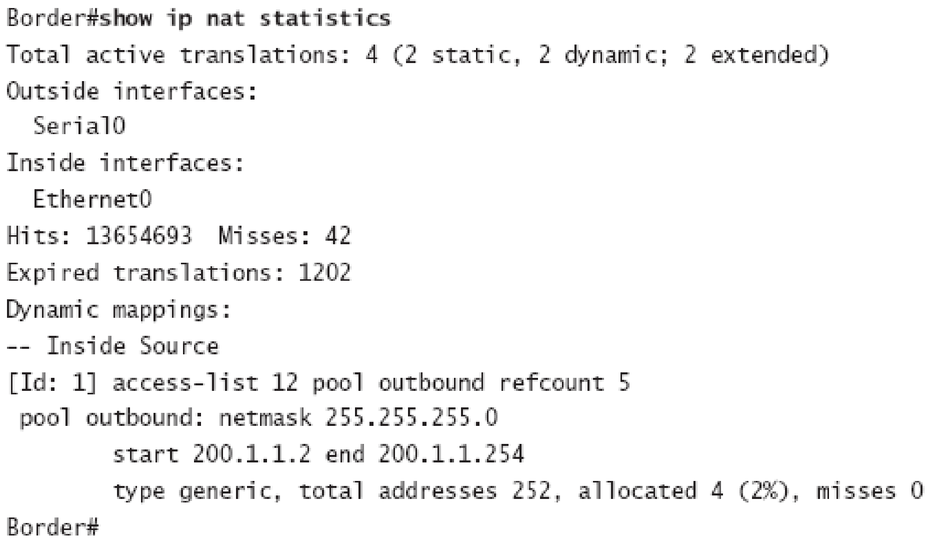
Adding the verbose keyword at the end of the command will display more information about each NAT table entry. These items include how long ago the entry was created, when it was last used, and how long before the entry will expire. The following is the output from adding the verbose keyword, shown in fig. 2.3:



```
Border#show ip nat translations verbose
Pro Inside global  Inside local   Outside local  Outside global
--- 200.1.1.25     10.1.1.25      ---            ---
    create 2d18h, use 2d18h, flags: static, use_count: 0
--- 200.1.1.26     10.1.1.26      ---                ---
    create 2d18h, use 2d18h, flags: static, use_count: 0
tcp 200.1.1.50:25  10.1.1.50:25   206.1.1.25:25  206.1.1.25:25
    create 05:53:05, use 05:53:05, left 18:06:54, flags: extended,
      ➥use_count: 0
tcp 200.1.1.51:514 10.1.1.51:514  155.1.9.6:1021 155.1.9.6:1021
    create 02:22:51, use 00:22:28, left 23:37:31, flags: extended,
      ➥use_count: 0
Border#
```

Figure 11.3: Detailed NAT Translation tables

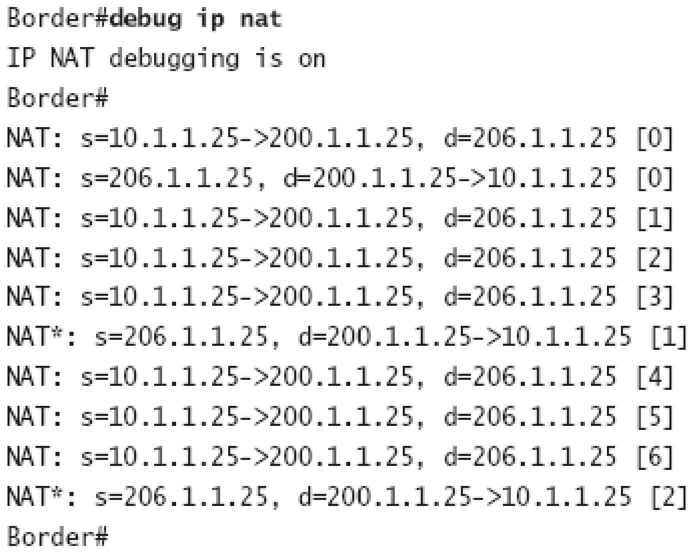
The second command is used to display the statistics and configuration information for NAT. The *show ip nat statistics* command displays the following information about the NAT table and statistics, as shown in figure 3.4:



```
Border#show ip nat statistics
Total active translations: 4 (2 static, 2 dynamic; 2 extended)
Outside interfaces:
  Serial0
Inside interfaces:
  Ethernet0
Hits: 13654693  Misses: 42
Expired translations: 1202
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 12 pool outbound refcount 5
 pool outbound: netmask 255.255.255.0
        start 200.1.1.2 end 200.1.1.254
        type generic, total addresses 252, allocated 4 (2%), misses 0
Border#
```

Figure 11.4: NAT statistics

The debug ip nat command is used to troubleshoot NAT problems on the router. In the following output you will notice that the inside local source address of 10.1.1.25, which gets translated to an inside global source address of 200.1.1.25, is sending a packet to the destination address 206.1.1.25. An arrow (—>) symbol indicates that the packet was translated, and an asterisk (*) symbol indicates that the packet is traveling through the fast path. The first packet in a conversation will be processed through a process-switched or slow path, and additional packets will be able to be switched faster through the fast path. The following example shows the output from the debug ip nat command, as shown in figure 3.5:



```
Border#debug ip nat
IP NAT debugging is on
Border#
NAT: s=10.1.1.25->200.1.1.25, d=206.1.1.25 [0]
NAT: s=206.1.1.25, d=200.1.1.25->10.1.1.25 [0]
NAT: s=10.1.1.25->200.1.1.25, d=206.1.1.25 [1]
NAT: s=10.1.1.25->200.1.1.25, d=206.1.1.25 [2]
NAT: s=10.1.1.25->200.1.1.25, d=206.1.1.25 [3]
NAT*: s=206.1.1.25, d=200.1.1.25->10.1.1.25 [1]
NAT: s=10.1.1.25->200.1.1.25, d=206.1.1.25 [4]
NAT: s=10.1.1.25->200.1.1.25, d=206.1.1.25 [5]
NAT: s=10.1.1.25->200.1.1.25, d=206.1.1.25 [6]
NAT*: s=206.1.1.25, d=200.1.1.25->10.1.1.25 [2]
Border#
```

Figure 11.5: Debugging NAT

Once debugging is enabled, it remains in effect until you turn it off with the no debug ip nat command; to turn off all debugging, use the undebug all command.

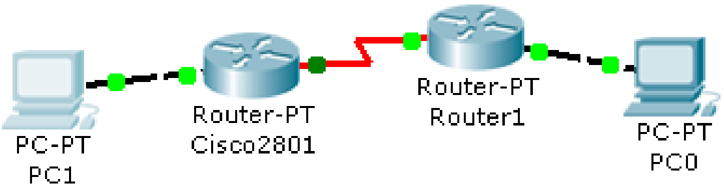


Figure 11.6: NAT implementation scenario

Table 11.1 *IP addresses for devices*

IP addresses for cisco2801	IP addresses for router1	IP addresses for PCs
172.18.1.2 /31 on serial interface	172.18.1.1 on serial interface	PC0 10.1.1.2/16
10.2.1.1 /16 on Ethernet interface	10.1.1.1 on Ethernet interface	PC1 10.2.1.6 /16

```
Cisco2801(config)#int s0/1/0
Cisco2801(config-if)# ip address 172.18.1.2 255.255.255.252
cisco2801(config-if)#ip nat outside
Cisco2801(config-if)#exit
cisco2801(config)#int fa 0/0
cisco2801(config-if)# ip address 10.2.1.1  255.255.0.0
cisco2801(config-if)#ip nat inside
cisco2801(config-if)#exit
```

<do configurations for router 1 and remember to add static routes on both routers. Static route for cisco2801 is given below>

```
cisco2801(config)#ip route 10.1.0.0  255.255.0.0 172.18.1.1
cisco2801(config)#ip nat inside source static 10.2.1.6 172.18.1.2
cisco2801(config)#exit
cisco2801#
cisco2801#sh ip nat trans
Pro  Inside global     Inside local       Outside local      Outside global
---  172.18.1.2        10.2.1.6           ---                ---
cisco2801(config)#int s 0/1/0
cisco2801(config-if)#ip nat outside
cisco2801(config-if)#exit
cisco2801(config)#ip nat outside source static 10.1.0.0 10.2.1.5
cisco2801(config)#exit
cisco2801#sh ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 172.18.1.2:5      10.2.1.6:5         10.1.1.2:5         10.1.1.2:5
icmp 172.18.1.2:6      10.2.1.6:6         10.1.1.2:6         10.1.1.2:6
icmp 172.18.1.2:7      10.2.1.6:7         10.1.1.2:7         10.1.1.2:7
icmp 172.18.1.2:8      10.2.1.6:8         10.1.1.2:8         10.1.1.2:8
---  172.18.1.2        10.2.1.6           ---                ---
---  ---               ---                10.2.1.5           10.1.0.0
```

NAT POOL

<in continuation to previous EXERCISE>

```
cisco2801(config)#no ip nat outside source static 10.1.0.0 10.2.1.5
cisco2801(config)#ip nat pool TO_INNER 10.2.1.7 10.2.1.22 netmask
255.255.0.0
```

EXERCISES

1. Use the NAT pool command to configure the above given NAT pool. Generate a request from the outside host to access internal network (through ping command to internal host). Now record the IP NAT translations and IP NAT statistics.

____

____

____

____

____

____

____

____

____

____

2. Use a standard access-list to allow all hosts from the network 10.2.1.0 to access the external network, giving them IPs from the pool 172.18.1.10 to 172.18.1.15. Write configuration commands for the same and show ip nat translation table.

____

____

____

____

____

____

____

____

____

____

____

____

____

____

Lab Session 12

OBJECT

Configure BGP on Cisco routers

THEORY

BGP is a protocol used by ISPs in order to achieve communication between different autonomous systems.

An autonomous system is a set of routers under one administrative domain, or those following one routing policy.

Since internet is heterogeneous network, there cannot be one interior routing protocol for connecting all networks. In any situation when a number of autonomous systems (different IGPs) have to be connected, or routing policies differ significantly, a common interior routing protocol cannot be implemented. In such cases, BGP proved to be a better alternative.
BGP runs over TCP port 179. It is a distance vector protocol in that it relies on the downstream neighbors' advertised routes to calculate a router's own routing table and then passes the results to the upstream routers, however the distance is not signified through a single number . Instead, BGP uses a list of AS numbers through which a packet must pass to reach its destination. This describes the full path of a packet. For this reason BGP is called a path vector routing protocol.

BGP, unlike IGPs, does not show details of topology inside an AS and thus maintains a routing table different from IGPs.

BGP comes in two versions IBGP and EBGP, depending on whether it is peering with a neighboring router inside its AS or in an external AS.

BGP shares a number of path attributes with its neighbors while advertising a route. These form the metric for BGP. There are four types of path attribute:

- Well known- Implemented by all BGP router.
  - Mandatory- must be carried with all BGP updates.
  - Discretionary- may not be carried in all updates.
- Optional- May not be covered in all BGP implementations.
  - Transitive- must be passed on to the upstream routers, if not implemented.
  - Non transitive- the carrying update may be dropped and the attributed may not be passed if not implemented.

<u>BGP path attributes</u>

	Attribute name	Class	Description
1.	ORIGIN	Well-known, mandatory	Specifies the origin of the routing update. Origin codes in the order of preference are IGP, EGP, Incomplete
2.	AS_PATH	Well-known,	Uses a sequence of AS numbers to describe the

		mandatory	inter-AS path. Shorter AS-path is preferred.
3.	NEXT_HOP	Well-known, mandatory	IP address of the next hop router on the path to the advertised destination. Lower IGP metric to next hop is preferred.
4.	LOCAL_PREF	Well-known, discretionary	Used only in updates between internal BGP peers. Communicates a router's degree of preference for an advertised router. For multiple routes to the same destination, one with highest local preference is taken.
5.	MULTI_EXIT_DISC	Optional, non-transitive	Allows an AS to inform other ASs about its preferred ingress points. For multiple routes to the same destination lower value of MED is preferred.
6.	ATOMIC_AGGREGATE and AGGREGATOR	Well-known, discretionary Optional-Transitive	Atomic aggregate is used to alert the downstream routers that a loss of path information has occurred due to summarization. Aggregator attribute tells where the aggregation was performed through AS No. and IP address of aggregator.
7.	COMMUNITY	Optional, Transitive	Identifies the destination AS as a member of some community that share one or more common properties.
8.	ORIGINATOR-ID, CLUSTER_LIST	Optional, non-transitive	Used to avoid routing loops

BGP decision process

Three information bases are involved in BGP routing. Adj_RIB_In; for received unprocessed routes, Local-RIB ; to form local routing table and Adj_RIB_Out; that contains routes to be advertised to peers.

The decision process entails three phases

1. Calculate the degree of preference for each feasible route.
2. Choose the best route and install in Local_RIB.
3. Add appropriate routes to Adj_RIB _Out

Configuring BGP

Create the following topology using 7200 series routers. Apply IP addresses to the routers' interfaces as shown in figure; also make necessary changes to the device names. Now configure AS for each router using the following command.

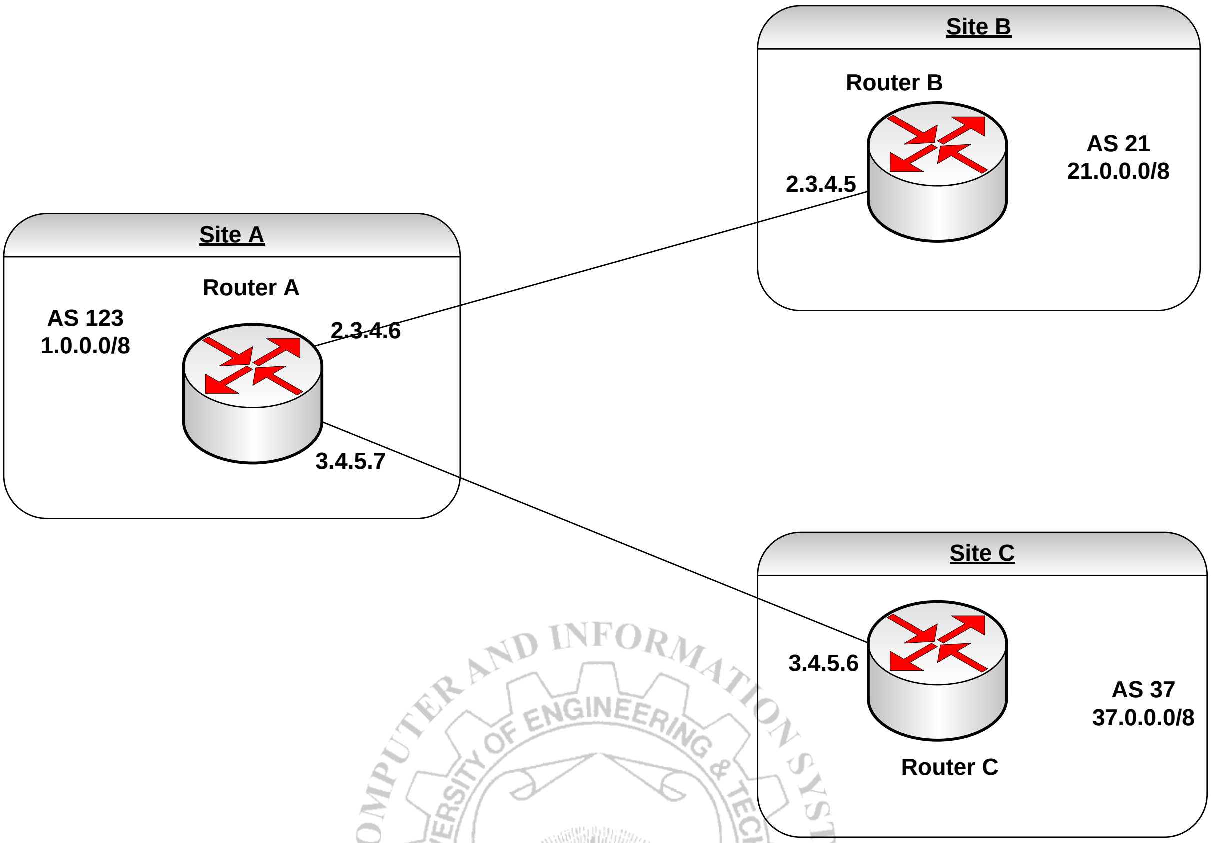


Figure 12.1 BGP Network

Configuring the basic BGP

Router A Configuration

```
router bgp 123  (this commands starts the BGP process. )
network 1.0.0.0
neighbor 2.3.4.5 remote-as 21
neighbor 3.4.5.6 remote-as 37
```

Router B Configuration

```
router bgp 21
network 21.0.0.0
neighbor 2.3.4.6 remote-as 123
```

Router C Configuration

```
router bgp 37
network 37.0.0.0
neighbor 3.4.5.7 remote-as 123
```

To temporarily disable BGP neighbor, use *neighbor ip-address shutdown command*

Monitoring and Troubleshooting the BGP

```
Show ip bgp
show ip bgp ip-prefix
show ip bgp summary
show ip bgp neighbors
show ip bgp neighbors ip-address
debug ip bgp events
```

EXERCISES

1. Add another AS – 40 in site 'D', to the given topology, containing network address 40.0.0.0/8 linked with both sites B and C. A common network with site B should be 4.2.2.0/30 and with site C 4.3.3.0/30. Write the configurations for the router D in this AS.

____

____

____

____

____

____

____

____

2. Write down the command to notice the BGP routing table of router D and note your observations in the space given below.

____

____

____

____

____

____

____

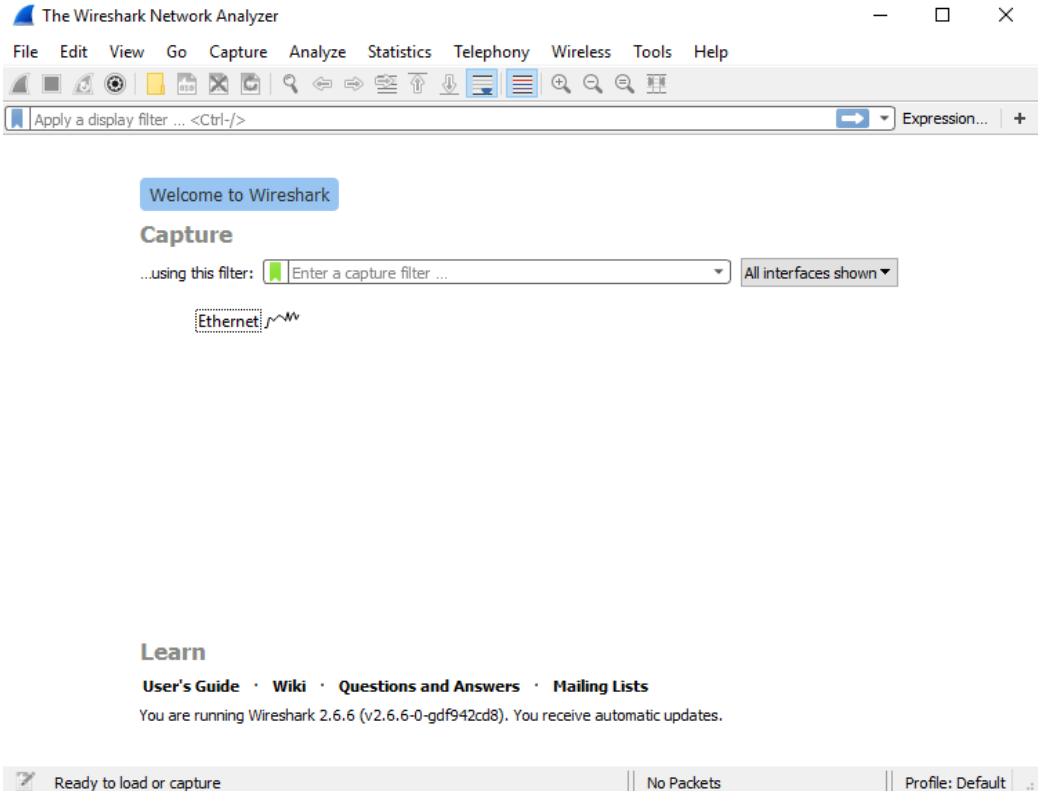
Lab Session 13

OBJECT

Examine packets of different protocols using Wireshark (Network Traffic Analysis and Filtering)

THEORY

This lab session introduces packet sniffer, Wireshark. Wireshark is a free open-source network protocol analyzer. It is used for network troubleshooting and communication protocol analysis. Wireshark captures network packets in real time and display them in human-readable format. It provides many advanced features including live capture and offline analysis, three-pane packet browser, coloring rules for analysis.

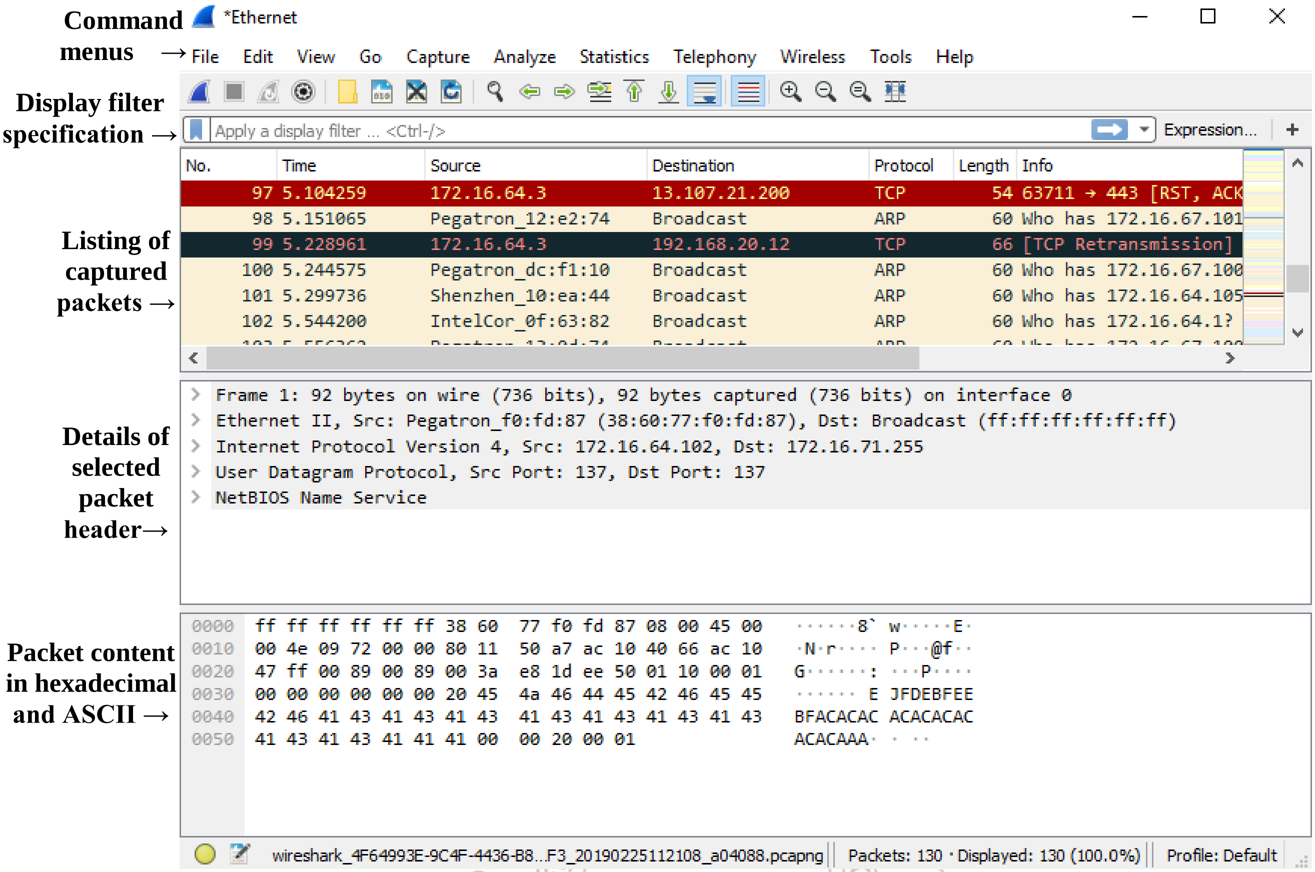


You are running Wireshark 2.6.6 (v2.6.6-0-gdf942cd8). You receive automatic updates.

Ready to load or capture | No Packets | Profile: Default

Figure 13.1: Initial Graphical User Interface of Wireshark

When you run the Wireshark program, the Wireshark graphic user interface will be shown as Figure 13.1. Currently, the program is not capturing the packets.
Then, you need to choose an interface. If you are running the Wireshark on your laptop, you need to select Wi-Fi interface. If you are at a desktop, you need to select the Ethernet interface being used. Note that there could be multiple interfaces. In general, you can select any interface but that does not mean that traffic will flow through that interface.


Command menus →
Display filter specification →
Listing of captured packets →
Details of selected packet header→
Packet content in hexadecimal and ASCII →
*Ethernet
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter … <Ctrl-/>
Expression…
+

No.	Time	Source	Destination	Protocol	Length	Info
97	5.104259	172.16.64.3	13.107.21.200	TCP	54	63711 → 443 [RST, ACK
98	5.151065	Pegatron_12:e2:74	Broadcast	ARP	60	Who has 172.16.67.101
99	5.228961	172.16.64.3	192.168.20.12	TCP	66	[TCP Retransmission]
100	5.244575	Pegatron_dc:f1:10	Broadcast	ARP	60	Who has 172.16.67.100
101	5.299736	Shenzhen_10:ea:44	Broadcast	ARP	60	Who has 172.16.64.105
102	5.544200	IntelCor_0f:63:82	Broadcast	ARP	60	Who has 172.16.64.1?

```
> Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
> Ethernet II, Src: Pegatron_f0:fd:87 (38:60:77:f0:fd:87), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 172.16.64.102, Dst: 172.16.71.255
> User Datagram Protocol, Src Port: 137, Dst Port: 137
> NetBIOS Name Service
```

```
0000  ff ff ff ff ff ff 38 60  77 f0 fd 87 08 00 45 00   ······8` w·····E·
0010  00 4e 09 72 00 00 80 11  50 a7 ac 10 40 66 ac 10   ·N·r···· P···@f··
0020  47 ff 00 89 00 89 00 3a  e8 1d ee 50 01 10 00 01   G······: ···P····
0030  00 00 00 00 00 00 20 45  4a 46 44 45 42 46 45 45   ······ E JFDEBFEE
0040  42 46 41 43 41 43 41 43  41 43 41 43 41 43 41 43   BFACACAC ACACACAC
0050  41 43 41 43 41 41 41 00  00 20 00 01               ACACAAA· · ··
```

wireshark_4F64993E-9C4F-4436-B8...F3_20190225112108_a04088.pcapng | Packets: 130 · Displayed: 130 (100.0%) | Profile: Default

Figure 13.2: Major components of Wireshark GUI

The Wireshark interface has five major components:

The **command menus** are standard pull-down menus located at the top of the window. Of interest to us now is the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.

The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

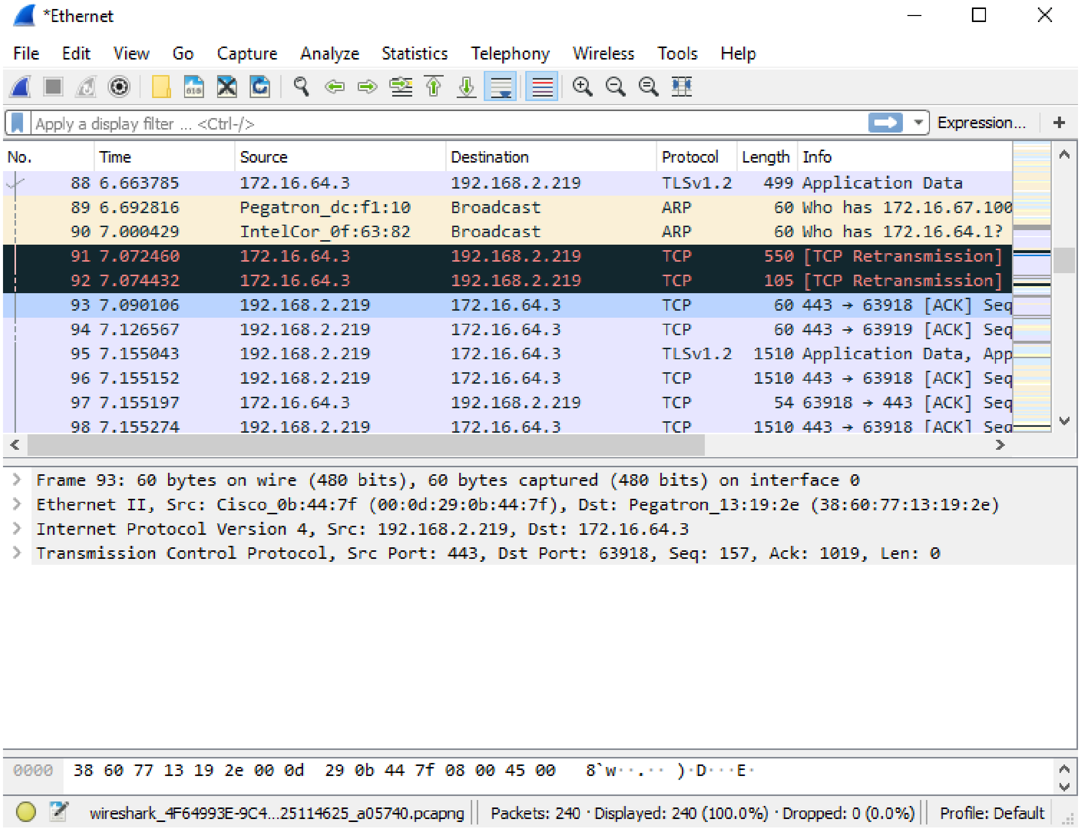
The **packet-header details window** provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet's one-line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the right-pointing or down-pointing arrowhead to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.

The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

Towards the top of the Wireshark graphical user interface, is the **packet display filter field**, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows).

Capture Packets

1- Double click on the available interface to start capturing.
2- Startup the browser.
3- In your browser, go to Google homepage by typing www.google.com.
4- After your browser has displayed the http://www.google.com page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture see image below:



```
No.  Time       Source             Destination    Protocol  Length Info
  88 6.663785   172.16.64.3        192.168.2.219  TLSv1.2     499 Application Data
  89 6.692816   Pegatron_dc:f1:10  Broadcast      ARP          60 Who has 172.16.67.100
  90 7.000429   IntelCor_0f:63:82  Broadcast      ARP          60 Who has 172.16.64.1?
  91 7.072460   172.16.64.3        192.168.2.219  TCP         550 [TCP Retransmission]
  92 7.074432   172.16.64.3        192.168.2.219  TCP         105 [TCP Retransmission]
  93 7.090106   192.168.2.219      172.16.64.3    TCP          60 443 → 63918 [ACK] Seq
  94 7.126567   192.168.2.219      172.16.64.3    TCP          60 443 → 63919 [ACK] Seq
  95 7.155043   192.168.2.219      172.16.64.3    TLSv1.2    1510 Application Data, App
  96 7.155152   192.168.2.219      172.16.64.3    TCP        1510 443 → 63918 [ACK] Seq
  97 7.155197   172.16.64.3        192.168.2.219  TCP          54 63918 → 443 [ACK] Seq
  98 7.155274   192.168.2.219      172.16.64.3    TCP        1510 443 → 63918 [ACK] Seq
```

```
> Frame 93: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: Cisco_0b:44:7f (00:0d:29:0b:44:7f), Dst: Pegatron_13:19:2e (38:60:77:13:19:2e)
> Internet Protocol Version 4, Src: 192.168.2.219, Dst: 172.16.64.3
> Transmission Control Protocol, Src Port: 443, Dst Port: 63918, Seq: 157, Ack: 1019, Len: 0
```

```
0000  38 60 77 13 19 2e 00 0d  29 0b 44 7f 08 00 45 00   8`w··.·· )·D···E·
```

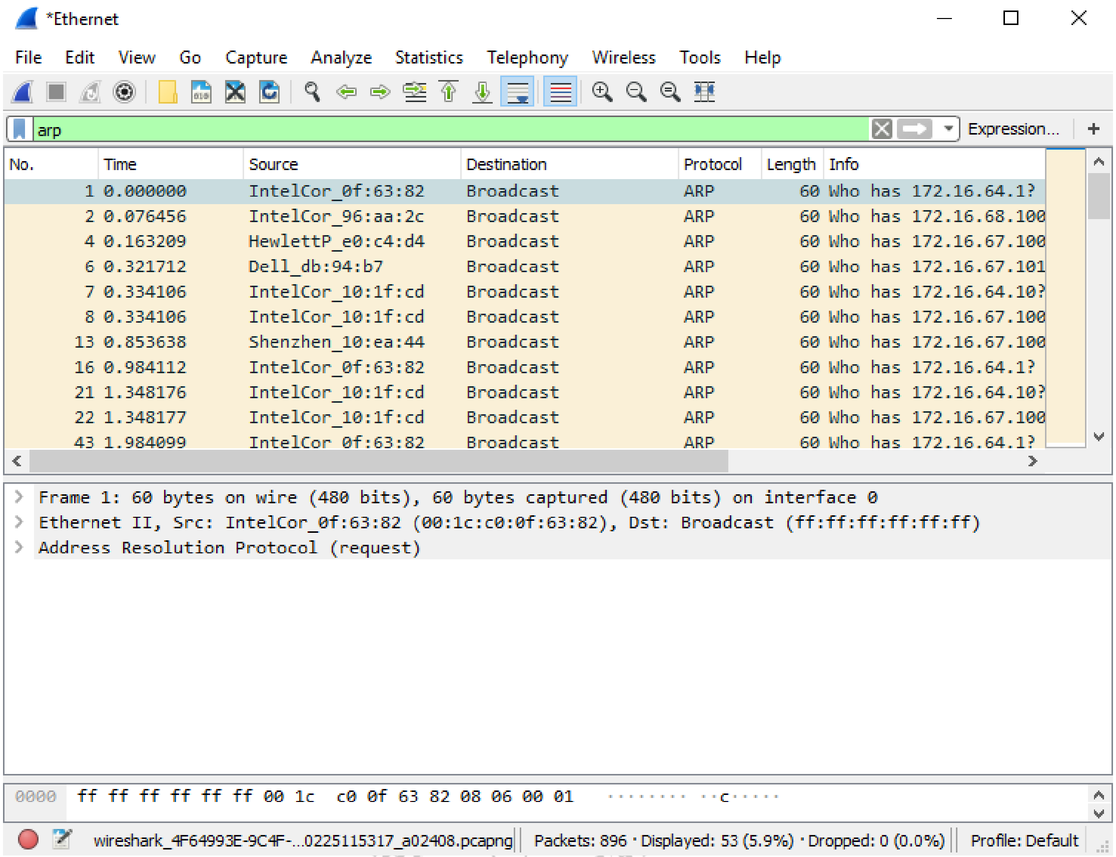
wireshark_4F64993E-9C4...25114625_a05740.pcapng | Packets: 240 · Displayed: 240 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

Figure 13.3: Capturing Packets in Wireshark

5- Color Coding: You’ll probably see packets highlighted in light purple, black and light yellow. Wireshark uses colors to help you identify the types of traffic at a glance. By default, light purple is TCP traffic, light yellow is ARP traffic and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

6- You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! Even though the only action you took was to open your browser, there are many other programs in your computer that communicate via the network in the background. To filter the connections to the ones we want to focus on, we have to use the filtering functionality of Wireshark.

7- Type "*name-of-protocol*" in the filtering field. An example with ARP is shown below:



arp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	IntelCor_0f:63:82	Broadcast	ARP	60	Who has 172.16.64.1?
2	0.076456	IntelCor_96:aa:2c	Broadcast	ARP	60	Who has 172.16.68.100
4	0.163209	HewlettP_e0:c4:d4	Broadcast	ARP	60	Who has 172.16.67.100
6	0.321712	Dell_db:94:b7	Broadcast	ARP	60	Who has 172.16.67.101
7	0.334106	IntelCor_10:1f:cd	Broadcast	ARP	60	Who has 172.16.64.10?
8	0.334106	IntelCor_10:1f:cd	Broadcast	ARP	60	Who has 172.16.67.100
13	0.853638	Shenzhen_10:ea:44	Broadcast	ARP	60	Who has 172.16.67.100
16	0.984112	IntelCor_0f:63:82	Broadcast	ARP	60	Who has 172.16.64.1?
21	1.348176	IntelCor_10:1f:cd	Broadcast	ARP	60	Who has 172.16.64.10?
22	1.348177	IntelCor_10:1f:cd	Broadcast	ARP	60	Who has 172.16.67.100
43	1.984099	IntelCor_0f:63:82	Broadcast	ARP	60	Who has 172.16.64.1?

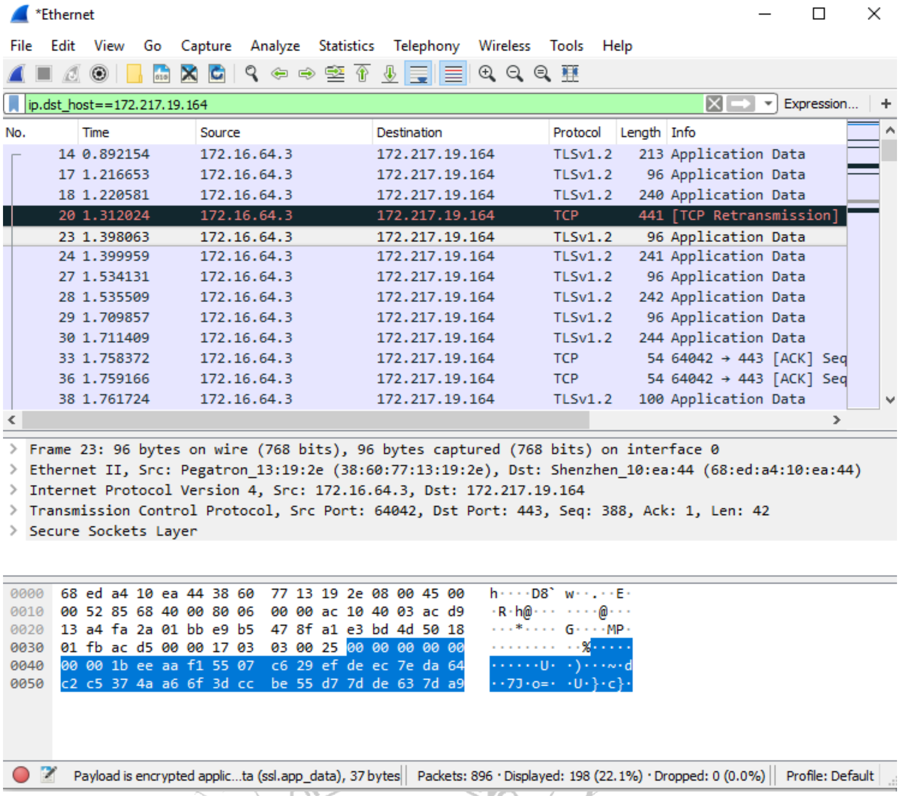
```
> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: IntelCor_0f:63:82 (00:1c:c0:0f:63:82), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 00 1c  c0 0f 63 82 08 06 00 01   ········ ··c·····
```

wireshark_4F64993E-9C4F-...0225115317_a02408.pcapng | Packets: 896 · Displayed: 53 (5.9%) · Dropped: 0 (0.0%) | Profile: Default

Figure 13.4: Interface with view of ARP packets only

8- To further filter packets in Wireshark, we need to use a more precise filter. By setting the ip.dst_host==172.217.19.64 (i.e. IP address of www.google.com), we are restricting the view to packets that have destination of 172.217.19.64. Notice that we need two equal signs to perform the match "==" not just one. See the screenshot below:


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
+

```
ip.dst_host==172.217.19.164
```

```
No.   Time      Source       Destination     Protocol  Length Info
   14 0.892154  172.16.64.3  172.217.19.164  TLSv1.2   213 Application Data
   17 1.216653  172.16.64.3  172.217.19.164  TLSv1.2    96 Application Data
   18 1.220581  172.16.64.3  172.217.19.164  TLSv1.2   240 Application Data
   20 1.312024  172.16.64.3  172.217.19.164  TCP       441 [TCP Retransmission]
   23 1.398063  172.16.64.3  172.217.19.164  TLSv1.2    96 Application Data
   24 1.399959  172.16.64.3  172.217.19.164  TLSv1.2   241 Application Data
   27 1.534131  172.16.64.3  172.217.19.164  TLSv1.2    96 Application Data
   28 1.535509  172.16.64.3  172.217.19.164  TLSv1.2   242 Application Data
   29 1.709857  172.16.64.3  172.217.19.164  TLSv1.2    96 Application Data
   30 1.711409  172.16.64.3  172.217.19.164  TLSv1.2   244 Application Data
   33 1.758372  172.16.64.3  172.217.19.164  TCP        54 64042 → 443 [ACK] Seq
   36 1.759166  172.16.64.3  172.217.19.164  TCP        54 64042 → 443 [ACK] Seq
   38 1.761724  172.16.64.3  172.217.19.164  TLSv1.2   100 Application Data
```

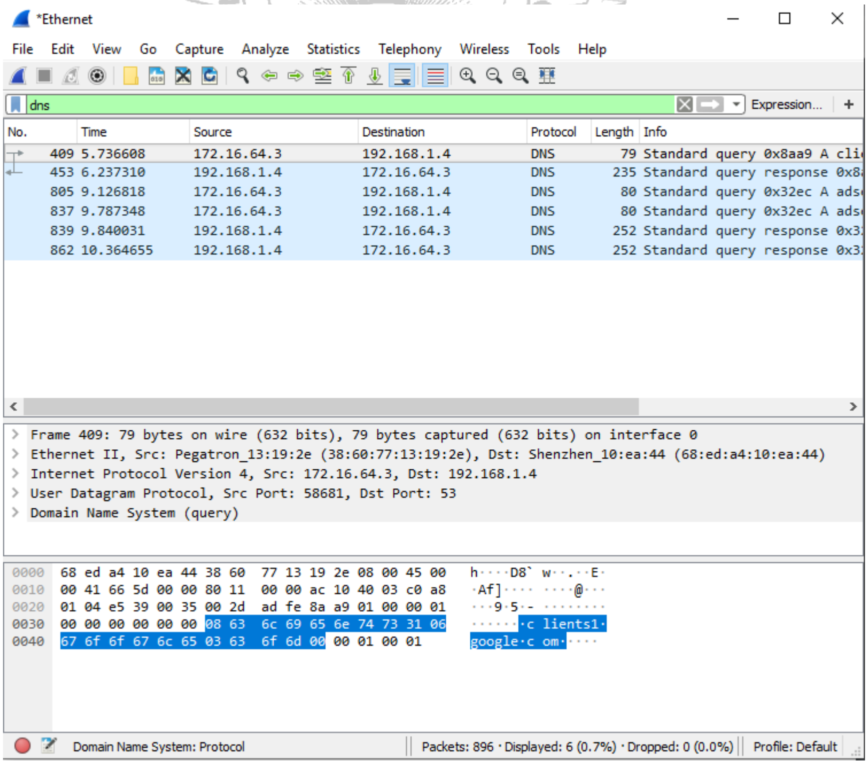
```
> Frame 23: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface 0
> Ethernet II, Src: Pegatron_13:19:2e (38:60:77:13:19:2e), Dst: Shenzhen_10:ea:44 (68:ed:a4:10:ea:44)
> Internet Protocol Version 4, Src: 172.16.64.3, Dst: 172.217.19.164
> Transmission Control Protocol, Src Port: 64042, Dst Port: 443, Seq: 388, Ack: 1, Len: 42
> Secure Sockets Layer
```

```
0000  68 ed a4 10 ea 44 38 60  77 13 19 2e 08 00 45 00   h····D8` w··.··E·
0010  00 52 85 68 40 00 80 06  00 00 ac 10 40 03 ac d9   ·R·h@··· ····@···
0020  13 a4 fa 2a 01 bb e9 b5  47 8f a1 e3 bd 4d 50 18   ···*···· G····MP·
0030  01 fb ac d5 00 00 17 03  03 00 25 00 00 00 00 00   ········ ··%·····
0040  00 00 1b ee aa f1 55 07  c6 29 ef de ec 7e da 64   ······U· ·)···~·d
0050  c2 c5 37 4a a6 6f 3d cc  be 55 d7 7d de 63 7d a9   ··7J·o=· ·U·}·c}·
```

```
Payload is encrypted applic…ta (ssl.app_data), 37 bytes | Packets: 896 · Displayed: 198 (22.1%) · Dropped: 0 (0.0%) | Profile: Default
```

Figure 13.4: Interface showing packets with destination 172.217.19.164 only

9- Now, we can try another protocol. Let's use Domain Name System (DNS) protocol as an example here.


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
+

```
dns
```

```
No.   Time       Source       Destination  Protocol  Length Info
  409 5.736608   172.16.64.3  192.168.1.4  DNS        79 Standard query 0x8aa9 A cli
  453 6.237310   192.168.1.4  172.16.64.3  DNS       235 Standard query response 0x8
  805 9.126818   172.16.64.3  192.168.1.4  DNS        80 Standard query 0x32ec A ads
  837 9.787348   172.16.64.3  192.168.1.4  DNS        80 Standard query 0x32ec A ads
  839 9.840031   192.168.1.4  172.16.64.3  DNS       252 Standard query response 0x3
  862 10.364655  192.168.1.4  172.16.64.3  DNS       252 Standard query response 0x3
```

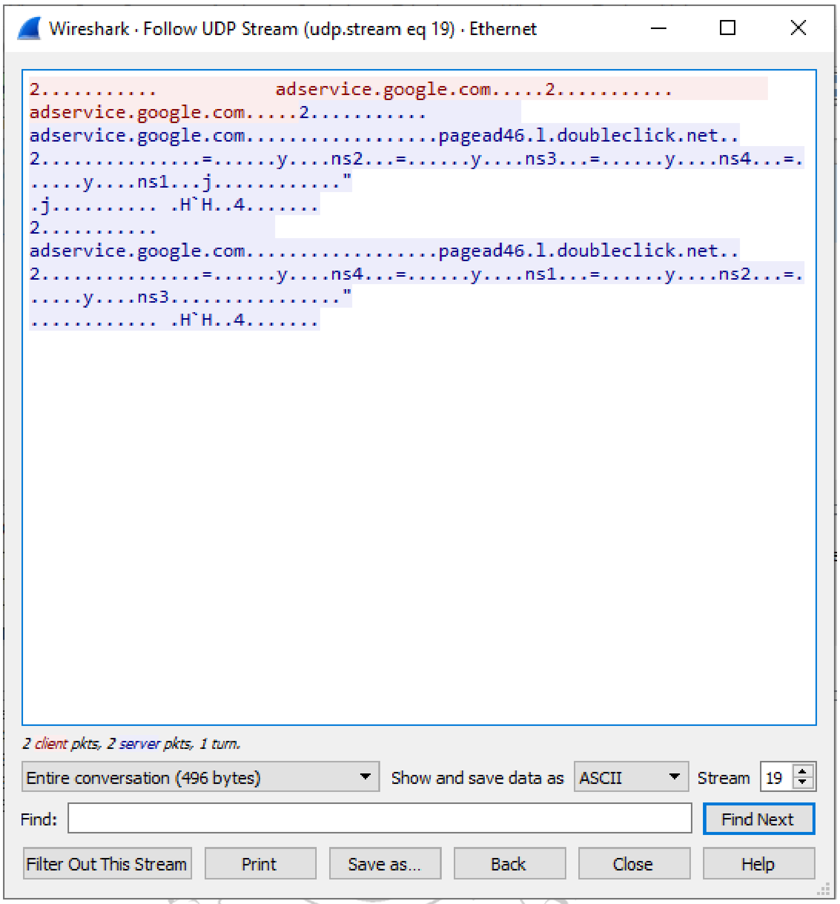
```
> Frame 409: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface 0
> Ethernet II, Src: Pegatron_13:19:2e (38:60:77:13:19:2e), Dst: Shenzhen_10:ea:44 (68:ed:a4:10:ea:44)
> Internet Protocol Version 4, Src: 172.16.64.3, Dst: 192.168.1.4
> User Datagram Protocol, Src Port: 58681, Dst Port: 53
> Domain Name System (query)
```

```
0000  68 ed a4 10 ea 44 38 60  77 13 19 2e 08 00 45 00   h····D8` w··.··E·
0010  00 41 66 5d 00 00 80 11  00 00 ac 10 40 03 c0 a8   ·Af]···· ····@···
0020  01 04 e5 39 00 35 00 2d  ad fe 8a a9 01 00 00 01   ···9·5·- ········
0030  00 00 00 00 00 00 08 63  6c 69 65 6e 74 73 31 06   ·······c lients1·
0040  67 6f 6f 67 6c 65 03 63  6f 6d 00 00 01 00 01      google·c om·····
```

```
Domain Name System: Protocol | Packets: 896 · Displayed: 6 (0.7%) · Dropped: 0 (0.0%) | Profile: Default
```

Figure 13.5: Interface showing packets of DNS protocol

10- Let’s try now to find out what these packets contain by following one of the conversations (also called network flows), select one of the packets and press the right mouse click. Select Flows -> UDP stream.



Wireshark · Follow UDP Stream (udp.stream eq 19) · Ethernet

```
2...........              adservice.google.com.....2...........
adservice.google.com.....2...........
adservice.google.com.................pagead46.l.doubleclick.net..
2..............=......y....ns2...=......y....ns3...=......y....ns4...=.
.....y....ns1...j............"
.j.......... .H`H..4.......
2...........
adservice.google.com.................pagead46.l.doubleclick.net..
2..............=......y....ns4...=......y....ns1...=......y....ns2...=.
.....y....ns3................"
............ .H`H..4.......
```

2 client pkts, 2 server pkts, 1 turn.

Entire conversation (496 bytes) Show and save data as ASCII Stream 19

Figure 13.6: Follow UDP Stream window

EXERCISE

1. Repeat steps 9 and 10 for other protocols in your capture. Attach screenshots for each protocol.

____

____

____

____

____

____

____

Attach screenshots here

Lab Session 14

OBJECT

Configure DHCP server & Helper address feature in Cisco router

THEORY

Cisco devices running Cisco software include Dynamic Host Configuration Protocol (DHCP) server. The Cisco IOS DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the device to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the Domain Name System (DNS) server and the default device. Consider the scenario given below. Let us configure a DHCP server for this simple network.

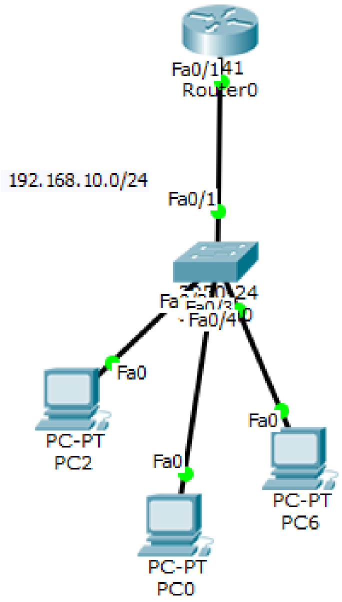
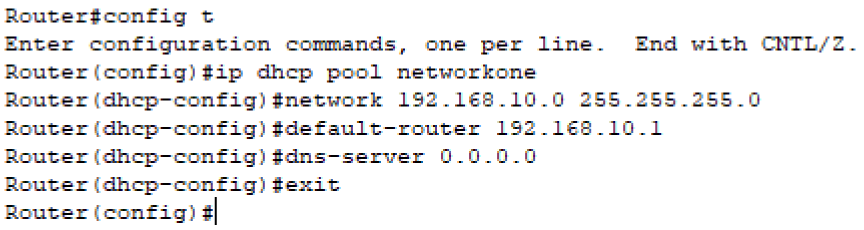


Figure 14.1: Scenario for configuration of DHCP Server

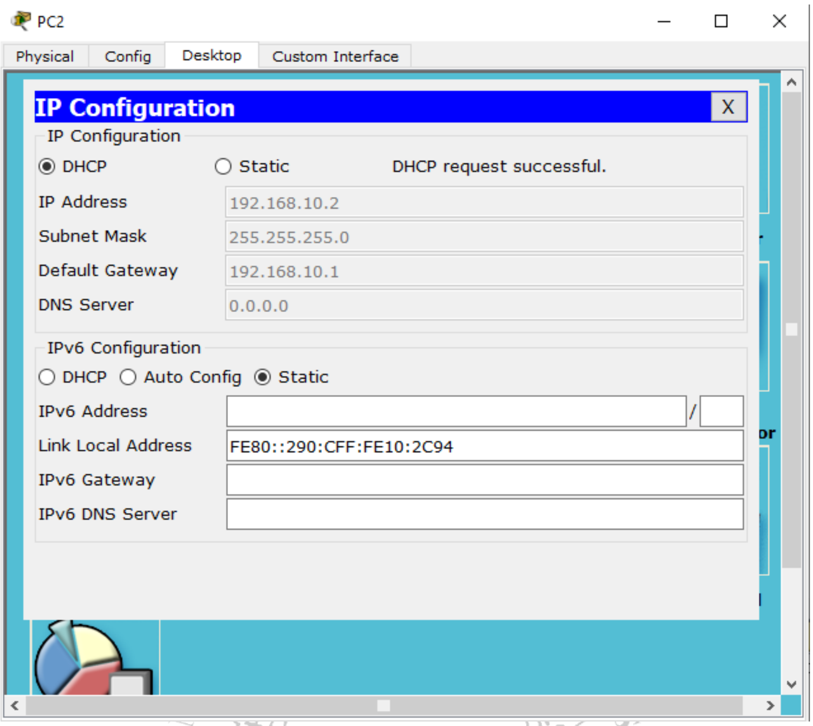
PROCEDURE

1- Make the connections as shown in the figure above.
2- Assign IP address *192.168.10.1/24* to the interface attached with the router.
3- Configure a DHCP address pool, default router and DNS server.



```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip dhcp pool networkone
Router(dhcp-config)#network 192.168.10.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.10.1
Router(dhcp-config)#dns-server 0.0.0.0
Router(dhcp-config)#exit
Router(config)#
```

4- Double click the PC. Go to Desktop -> IP configuration. Select the DHCP option. Repeat this step for all PCs.



IP Configuration

IP Configuration

● DHCP ○ Static DHCP request successful.

IP Address 192.168.10.2

Subnet Mask 255.255.255.0

Default Gateway 192.168.10.1

DNS Server 0.0.0.0

IPv6 Configuration

○ DHCP ○ Auto Config ● Static

IPv6 Address /

Link Local Address FE80::290:CFF:FE10:2C94

IPv6 Gateway

IPv6 DNS Server

Figure 14.2: DHCP request sent successfully (Dynamic IP allotted)

5- You can see the IP addresses being assigned to the PC dynamically.

Excluding IP Addresses

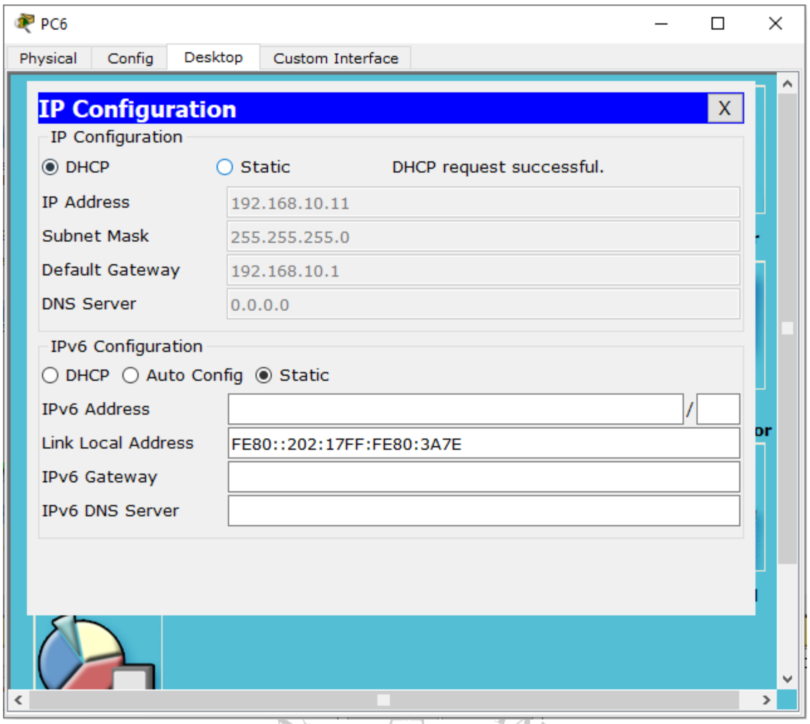
The IP address configured on a device interface is automatically excluded from the DHCP address pool. The DHCP server assumes that all other IP addresses in a DHCP address pool subnet are available for assigning to DHCP clients. You must exclude addresses from the pool if the DHCP server does not allocate those IP addresses to DHCP clients.

ip dhcp excluded-address *low-address* [*high-address*]

Example:

```
Router(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.10
```

The above command specifies the IP addresses that the DHCP server should not assign to DHCP clients.



IP Configuration

IP Configuration

● DHCP ○ Static DHCP request successful.

IP Address: 192.168.10.11

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.10.1

DNS Server: 0.0.0.0

IPv6 Configuration

○ DHCP ○ Auto Config ● Static

IPv6 Address: /

Link Local Address: FE80::202:17FF:FE80:3A7E

IPv6 Gateway:

IPv6 DNS Server:

Figure 14.3: IP address now assigned from 192.168.10.11

Configure DHCP IP Helper Address

DHCP IP Helper addresses are IP addresses configured on a routed interface such as a VLAN Interface or a routers Ethernet interface that allows that specific device to act as a “middle man” which forwards BOOTP (Broadcast) DHCP request it receives on an interface to the DHCP server specified by the IP Helper address via unicast.

Let’s extend the above figure into the scenario given below:

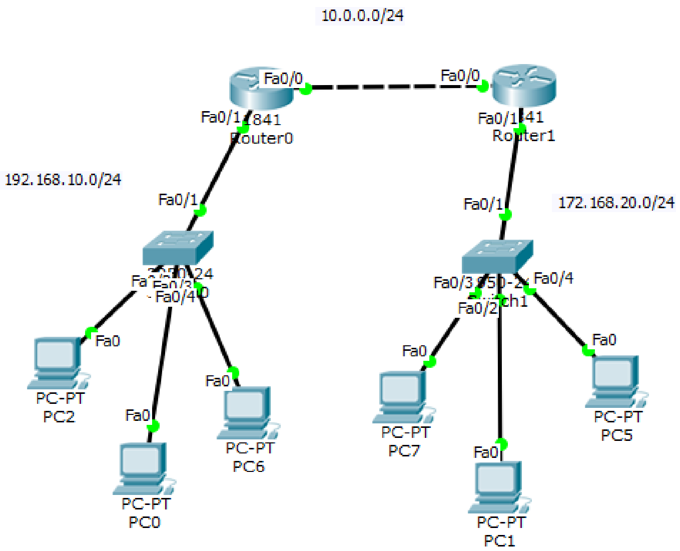
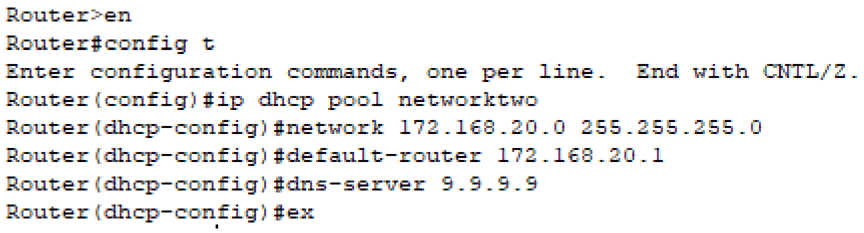


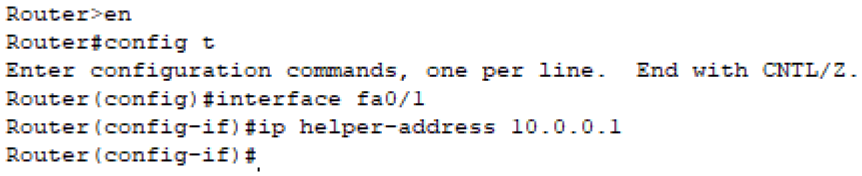
Figure 14.4: IP address now assigned from 192.168.10.11

1- Configure the IP addresses for the interfaces shown in the figure.
2- Configure DHCP pool, default-router and DNS server for the network *172.168.20.0/24*



```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip dhcp pool networktwo
Router(dhcp-config)#network 172.168.20.0 255.255.255.0
Router(dhcp-config)#default-router 172.168.20.1
Router(dhcp-config)#dns-server 9.9.9.9
Router(dhcp-config)#ex
```

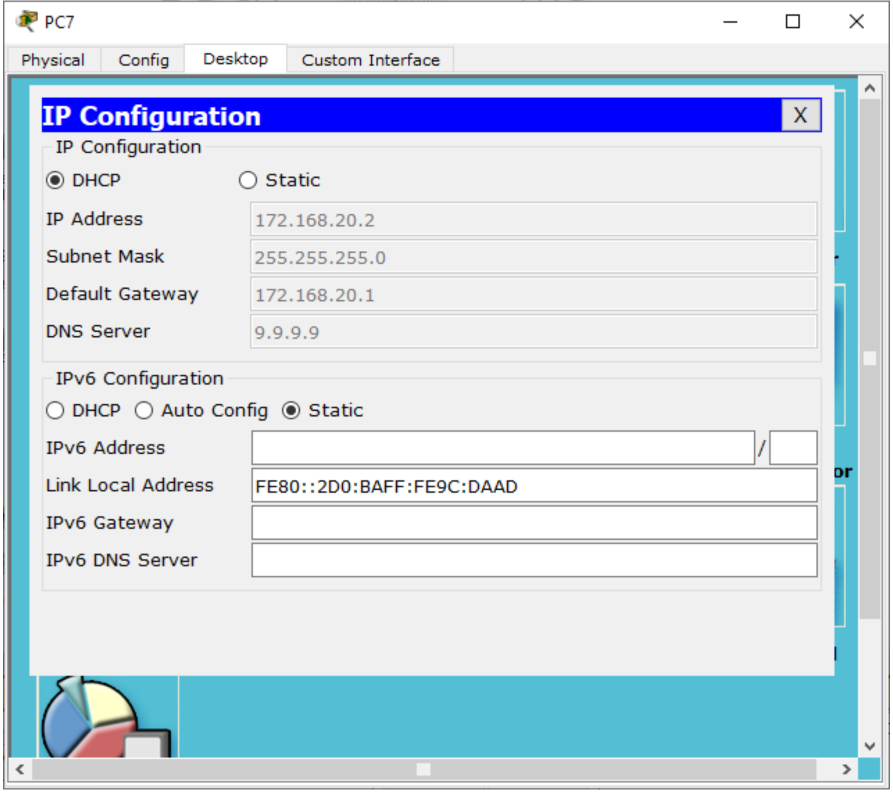
3- Now configure IP helper address on the connected router (i.e. router on the right side).
ip helper-address *ip-addr-of-router0-interface*



```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fa0/1
Router(config-if)#ip helper-address 10.0.0.1
Router(config-if)#
```

This command is executed in interface configuration mode to enable a Layer 3 interface to receive BOOTP DHCP Request and forward them to a specified DHCP server (i.e. router on the left side).

4- Configure RIP version 2 on both the routers.
5- Double click the PC on the right side. Go to Desktop -> IP configuration. Select the DHCP option. Repeat this step for all PCs.



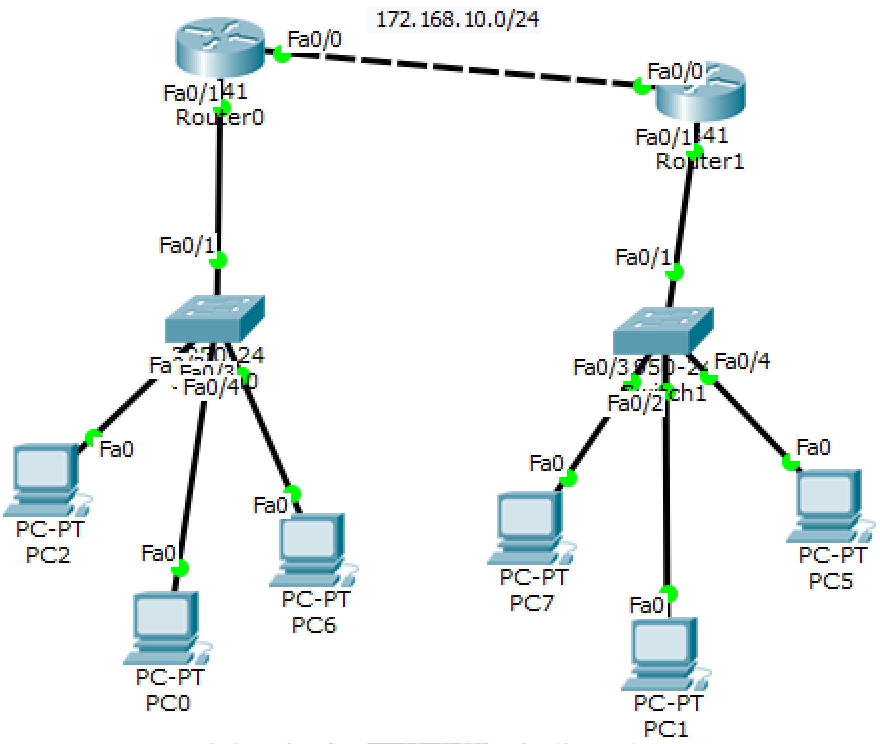
IP Configuration
● DHCP ○ Static
IP Address: 172.168.20.2
Subnet Mask: 255.255.255.0
Default Gateway: 172.168.20.1
DNS Server: 9.9.9.9

IPv6 Configuration
○ DHCP ○ Auto Config ● Static
IPv6 Address: /
Link Local Address: FE80::2D0:BAFF:FE9C:DAAD
IPv6 Gateway:
IPv6 DNS Server:

Figure 14.5: IP address now assigned from pool *172.168.20.0/24*

EXERCISE

1. Configure DHCP server with pools 10.10.10.0/24 and 192.168.20.0/27. Exclude addresses from 10.10.10.1-10.10.10.5/24. Also configure IP helper address for the router connected.



APPENDIX A

IPV4 ADDRESSING, SUB-NETTING AND VLSM

IP ADDRESS & SUBNET MASK

An IP (Internet Protocol) address uniquely identifies a node or host connection to an IP network. System administrators or network designers assign IP addresses to nodes. IP addresses are configured by software and are not hardware specific. An IP address is a 32 bit binary number usually represented as four fields each representing 8 bit numbers in the range 0 to 255 (sometimes called octets) separated by decimal points.
For example: 150.215.17.9

It is sometimes useful to view the values in their binary form.

150.215.17.9
10010110.11010111.00010001.00001001

An IP address consists of two parts, one identifying the network and one identifying the node. The class of the address determines which part belongs to the network address which part belongs to the node address.

An IP address has two components, the network address and the host address. A subnet mask separates the IP address into the network and host addresses (<network><host>)
A Subnet mask is a 32-bit number that masks an IP address, and divides the IP address into network address and host address. Subnet Mask is made by setting network bits to all "1"s and setting host bits to all "0"s.

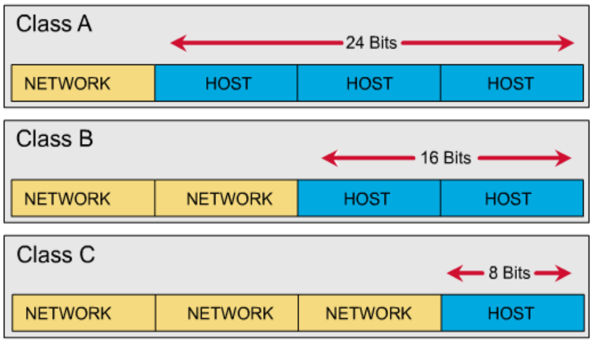


Figure A.1: Class A, B and C

CLASSFUL ADDRESSING

IPv4 addressing used the concept of classes. This architecture is called classful addressing. The address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the address space.

We can find the class of an address when given the address in binary notation or dotted-decimal notation. If the address is given in binary notation, the first few bits can immediately tell us the class

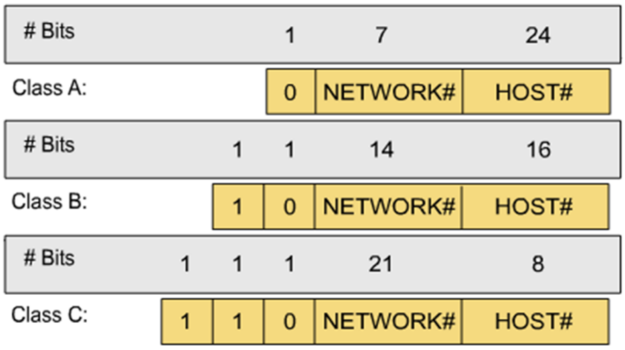
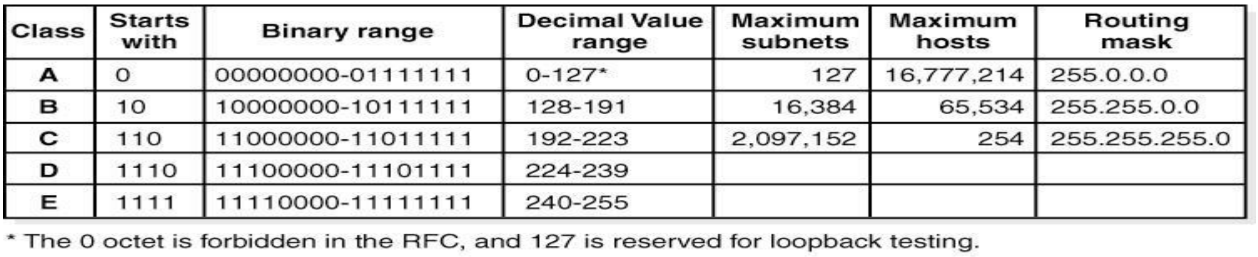

# Bits 1 7 24
Class A: 0 NETWORK# HOST#
# Bits 1 1 14 16
Class B: 1 0 NETWORK# HOST#
# Bits 1 1 1 21 8
Class C: 1 1 0 NETWORK# HOST#

Figure A.2: Network and Host bits

of the address. If the address is given in decimal-dotted notation, the first byte defines the class.



Class	Starts with	Binary range	Decimal Value range	Maximum subnets	Maximum hosts	Routing mask
A	0	00000000-01111111	0-127*	127	16,777,214	255.0.0.0
B	10	10000000-10111111	128-191	16,384	65,534	255.255.0.0
C	110	11000000-11011111	192-223	2,097,152	254	255.255.255.0
D	1110	11100000-11101111	224-239			
E	1111	11110000-11111111	240-255			

* The 0 octet is forbidden in the RFC, and 127 is reserved for loopback testing.

Network & Broadcast Addresses

- An IP address such as 176.10.0.0 that has all binary 0s in the host bit positions is reserved for the network address.
- An IP address such as 176.10.255.255 that has all binary 1s in the host bit positions is reserved for the broadcast address.

SUB-NETTING

To create a subnet address, a network administrator borrows bits from the original host portion and designates them as the subnet field.

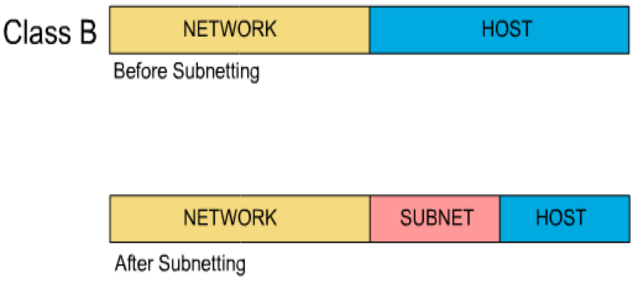
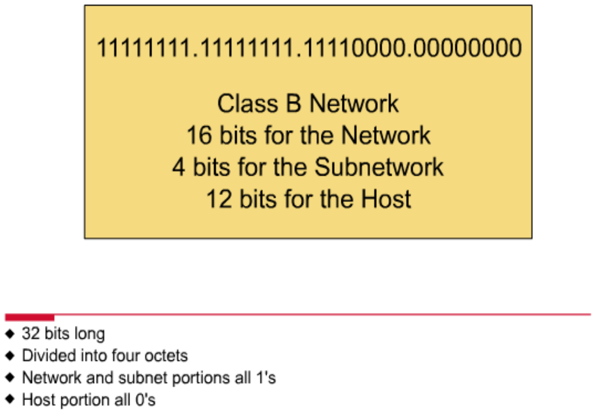


Figure A.3: Sub-netting

Consider the following example:



11111111.11111111.11110000.00000000

Class B Network
16 bits for the Network
4 bits for the Subnetwork
12 bits for the Host

- 32 bits long
- Divided into four octets
- Network and subnet portions all 1's
- Host portion all 0's

Figure A.4: Sub-netting of Class B network

Sub-netting Class C Addresses

In a Class C address, only 8 bits are available for defining the hosts. Remember that subnet bits start at the left and go to the right, without skipping bits. This means that the only Class C subnet masks can be the following:

Binary (4thOctect)	Decimal (4thOctect)	CIDR (Classless Inter-Domain Routing) or slash notation
10000000	128	/25
11000000	192	/26
11100000	224	/27
11110000	240	/28
11111000	248	/29
11111100	252	/30

Now determine the following:

How many subnets? 2^x = number of subnets. *x* is the number of masked bits, or the 1s. For example, in 11000000, the number of ones gives us 2^2 subnets. In this example, there are 4 subnets.

How many hosts per subnet? 2^y– 2 = number of hosts per subnet. *y* is the number of unmasked bits, or the 0s. For example, in 11000000, the number of zeros gives us 2^6 – 2 hosts. In this example, there are 62 hosts per subnet. You need to subtract two for the subnet address and the broadcast address, which are not valid hosts.

What are the valid subnets? 256 – Subnet mask = block size, or increment number. An example would be 256 – 192 = 64. The block size of a 192 mask is always 64. Start counting at zero in blocks of 64 until you reach the subnet mask value and these are your subnets: 0, 64, 128 and 192.

What's the broadcast address for each subnet? Since we counted our subnets in the last section as 0, 64, 128, and 192, the broadcast address is always the number right before the next subnet. For example, the 0 subnet has a broadcast address of 63 because the next subnet is 64. The 64 subnet has a broadcast address of 127 because the next subnet is 128, etc. The broadcast of the last subnet is always 255 for Class C.

What are the valid hosts? Valid hosts are the numbers between the subnets, omitting all the 0s and all 1s. For example, if 64 is the subnet number and 127 is the broadcast address, then 65–126 is the valid host range—it's *always* the numbers between the subnet address and the broadcast address.

Sub-netting (Class B & A addresses) & VLSM

Sub-netting Class B Addresses

Binary (3rd and 4thOctet)	Decimal (All Octets)	CIDR (Classless Inter-Domain Routing) or slash notation
10000000 00000000	255.255.128.0	/17
11000000 00000000	255.255.192.0	/18
11100000 00000000	255.255.224.0	/19
11110000 00000000	255.255.240.0	/20
11111000 00000000	255.255.248.0	/21
11111100 00000000	255.255.252.0	/22
11111110 00000000	255.255.254.0	/23
11111111 00000000	255.255.255.0	/24
11111111 10000000	255.255.255.128	/25
11111111 11000000	255.255.255.192	/26
11111111 11100000	255.255.255.224	/27
11111111 11110000	255.255.255.240	/28
11111111 11111000	255.255.255.248	/29
11111111 11111100	255.255.255.252	/30

Then determine all the parameters discussed in Lab 01 in Sub-netting Class C Address section.

Sub-netting Class A Addresses

Note that the Class A addresses can be sub-netted in the same way as for Class C & B. However, in that case we have sub-netting possible in 3 octets as opposed to 1 or 2 subnets as in Class C or B respectively.

VARIABLE LENGTH SUBNET MASKING (VLSM)

Variable Length Subnet Masking (VLSM) is a way of further sub-netting a subnet. Using Variable Length Subnet Masking (VLSM) we can allocate IP addresses to the subnets by the exact need (*in the power f 2*).
Variable Length Subnet Masking (VLSM) allows us to use more than one subnet mask within the same network address space.
If we recollect from the previous lessons, we can divide a network only into subnets with equal number of IP addresses. Variable Length Subnet Masking (VLSM) allows creating subnets from a single network with unequal number of IP addresses.

Example: We want to divide 192.168.10.0, which is a Class C network, into four networks, each with unequal number of IP address requirements as shown below.
Subnet A: *126 IP Addresses*
Subnet B: 62 IP Addresses
Subnet C: 30 IP Addresses
Subnet D: 30 IP Addresses

Original Network (Network to be sub-netted) – 192.168.10.0/24

(VLSM) - First Division

Divide the two networks equally with 128 IPv4 addresses (126 usable IPv4 addresses) in each network using 255.255.255.128subnet mask (192.168.10.0/25).

We will get two subnets each with 128 IPv4 addresses (126 usable IPv4 addresses).

1) 192.168.10.0/25, which can be represented in binaries as below.

11000000.10101000.00001010.00000000
11111111.11111111.11111111.10000000

2) 192.168.10.128/25, which can be represented in binaries as below.

11000000.10101000.00001010.10000000
11111111.11111111.11111111.10000000

(VLSM)- Second Division

Divide second subnet (192.168.10.128/25) we got from the first division again into two Networks, each with 64 IP Addresses (62 usable IPv4 addresses) using 255.255.255.192 subnet mask.

We will get two subnets each with 64 IPv4 addresses (62 usable IPv4 addresses).

1) 192.168.10.128/26, which can be represented in binaries as below.

11000000.10101000.00001010.10000000
11111111.11111111.11111111.11000000

2) 192.168.10.192/26

11000000.10101000.00001010.11000000
11111111.11111111.11111111.11000000

(VLSM) - Third Division

Divide 192.168.10.192/26 Network again into two Networks, each with 32 IPv4 addresses (30 usable IPv4 addresses) using 255.255.255.224 subnet mask

We will get two subnets each with 32 IPv4 addresses (30 usable IPv4 addresses).

1) 192.168.10.192/27, which can be represented in binaries as below.

11000000.10101000.00001010.11000000
11111111.11111111.11111111.11100000

2) 192.168.10.224/27, which can be represented in binaries as below.

11000000.10101000.00001010.11100000
11111111.11111111.11111111.11100000

Now we have split the 192.168.10.0/24 network into four subnets using Variable Length Subnet Masking (VLSM), with unequal number of IPv4 addresses as shown below. Also note that when you divide a network using Variable Length Subnet Masking (VLSM), the subnet masks are also different.

1) 192.168.10.0 - 255.255.255.128 (126 (128-2) usable IPv4 addresses)
2) 192.168.10.128 - 255.255.255.192 (62 (64-2) usable IPv4 addresses)
3) 192.168.10.192 - 255.255.255.224 (30 (32-2) usable IPv4 addresses)
4) 192.168.10.224 - 255.255.255.224 (30 (32-2) usable IPv4 addresses)

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING

RUBRIC

Course Code: _______________ Week #: _______________ Lab #: _______________

Assigned task: ____

____

CRITERIA AND SCALES		
Criterion 1: Demonstration of Concepts		
0	1	2
The student failed to demonstrate a clear understanding of the assignment concepts.	The student demonstrated a clear understanding of some of the assignment concepts.	The student demonstrated a clear understanding of the assignment concepts.
Criterion 2: Accuracy		
0	1	2
The student misconfigured enough network settings that the lab computer couldn't function properly on the network.	The student configured enough network settings that the lab computer partially functioned on the network.	The student configured the network settings that the lab computer fully functioned on the network.
Criterion 3: Following Directions		
0	1	2
The student clearly failed to follow the verbal and written instructions to successfully complete the lab.	The student failed to follow the some of the verbal and written instructions to successfully complete all requirements of the lab	The student followed the verbal and written instructions to successfully complete requirements of the lab.
Criterion 4: Time Utilization		
0	1	2
The student failed to complete even part of the lab in the allotted amount of time.	The student failed to complete the entire lab in the allotted amount of time.	The student completed the lab in its entirety in the allotted amount of time.
Criterion 5: Interaction of the student with lab resources (lab personnel, participant students, equipment)		
0	1	2
The student took no notice of the lab resources	The student was aware of lab resources for a short period of time but was mostly unconcerned	The student effectively interacted with the lab resources

____
Teacher’s Signature

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING

RUBRIC

Course Code: _______________ Week #: _______________ Lab #: _______________

Assigned task: ____

____

CRITERIA AND SCALES		
Criterion 1: Demonstration of Concepts		
0	1	2
The student failed to demonstrate a clear understanding of the assignment concepts.	The student demonstrated a clear understanding of some of the assignment concepts.	The student demonstrated a clear understanding of the assignment concepts.
Criterion 2: Accuracy		
0	1	2
The student misconfigured enough network settings that the lab computer couldn't function properly on the network.	The student configured enough network settings that the lab computer partially functioned on the network.	The student configured the network settings that the lab computer fully functioned on the network.
Criterion 3: Following Directions		
0	1	2
The student clearly failed to follow the verbal and written instructions to successfully complete the lab.	The student failed to follow the some of the verbal and written instructions to successfully complete all requirements of the lab	The student followed the verbal and written instructions to successfully complete requirements of the lab.
Criterion 4: Time Utilization		
0	1	2
The student failed to complete even part of the lab in the allotted amount of time.	The student failed to complete the entire lab in the allotted amount of time.	The student completed the lab in its entirety in the allotted amount of time.
Criterion 5: Interaction of the student with lab resources (lab personnel, participant students, equipment)		
0	1	2
The student took no notice of the lab resources	The student was aware of lab resources for a short period of time but was mostly unconcerned	The student effectively interacted with the lab resources

____
Teacher’s Signature

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING

RUBRIC

Course Code: ________________ Week #: ________________ Lab #: ________________

Assigned task: ____

____

CRITERIA AND SCALES		
Criterion 1: Confidence with tool interface		
0	1	2
The student is unfamiliar with the tool	The student is familiar with the visible features of the tool	The student is familiar with the unexposed features of the tool
Criterion 2: Appraisal of lab assignment		
0	1	2
The student did not understand the lab task	The student understood the lab task but had problem with tool usage	The student initiated implementation of the lab task but could not complete it
Criterion 3: Answer to questions related to the tool		
0	1	2
The student has no idea about tool application	The student has vague idea about tool application	The student has a clear idea about tool application
Criterion 4: Answer to questions related to the lab task		
0	1	2
The student did not answer any question regarding lab task	The student answered a few questions regarding lab task	The student answered most of the questions regarding lab task
Criterion 5: How would you grade the interaction of the student with lab resources (lab personnel, participant students, equipment)?		
0	1	2
The student took no notice of the lab resources	The student was aware of lab resources for a short period of time but was mostly unconcerned	The student effectively interacted with the lab resources

____
Teacher's Signature

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING

RUBRIC

Course Code: _______________ Week #: _______________ Lab #: _______________

Assigned task: ____

____

CRITERIA AND SCALES		
Criterion 1: Confidence with tool interface		
0	1	2
The student is unfamiliar with the tool	The student is familiar with the visible features of the tool	The student is familiar with the unexposed features of the tool
Criterion 2: Appraisal of lab assignment		
0	1	2
The student did not understand the lab task	The student understood the lab task but had problem with tool usage	The student initiated implementation of the lab task but could not complete it
Criterion 3: Answer to questions related to the tool		
0	1	2
The student has no idea about tool application	The student has vague idea about tool application	The student has a clear idea about tool application
Criterion 4: Answer to questions related to the lab task		
0	1	2
The student did not answer any question regarding lab task	The student answered a few questions regarding lab task	The student answered most of the questions regarding lab task
Criterion 5: How would you grade the interaction of the student with lab resources (lab personnel, participant students, equipment)?		
0	1	2
The student took no notice of the lab resources	The student was aware of lab resources for a short period of time but was mostly unconcerned	The student effectively interacted with the lab resources

Teacher’s Signature